

Volume 6

Pages 1133 - 1396

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

BEFORE THE HONORABLE WILLIAM H. ALSUP

ORACLE AMERICA, INC.,)	
)	
Plaintiff,)	
)	
VS.)	No. C 10-3561 WHA
)	
GOOGLE, INC.,)	
)	
Defendant.)	San Francisco, California
)	April 23, 2012

TRANSCRIPT OF JURY TRIAL PROCEEDINGS

APPEARANCES:

For Plaintiff:	MORRISON & FOERSTER 755 Page Mill Road Palo Alto, California 94304
BY:	MICHAEL A. JACOBS, ESQUIRE KENNETH A. KUWAYTI, ESQUIRE MARC DAVID PETERS, ESQUIRE DANIEL P. MUINO, ESQUIRE
	BOIES, SCHILLER & FLEXNER 333 Main Street Armonk, New York 10504
BY:	DAVID BOIES, ESQUIRE ALANNA RUTHERFORD, ESQUIRE

(Appearances continued on next page)

Reported By: Katherine Powell Sullivan, RPR, CRR, CSR #5812
Debra L. Pas, RMR, CRR, CSR #11916
Official Reporters - U.S. District Court

Katherine Powell Sullivan, CSR, CRR, RPR
Debra L. Pas, CSR, CRR, RMR
Official Reporters - US District Court - 415-794-6659

APPEARANCES (CONTINUED):

For Plaintiff:

BOIES, SCHILLER & FLEXNER
1999 Harrison Street, Suite 900
Oakland, California 94612
BY: WILLIAM FRED NORTON, ESQUIRE
STEVEN C. HOLTZMAN, ESQUIRE

ORACLE AMERICA, INC.
500 Oracle Parkway
Redwood Shores, California 94065
BY: ANDREW C. TEMKIN, CORPORATE COUNSEL
DORIAN DALEY, GENERAL COUNSEL

For Defendant:

KEKER & VAN NEST
633 Battery Street
San Francisco, California 94111-1809
BY: ROBERT ADDY VAN NEST, ESQUIRE
CHRISTA MARTINE ANDERSON, ESQUIRE
DANIEL PURCELL, ESQUIRE
MICHAEL S. KWUN, ESQUIRE

KING & SPALDING LLP
1185 Avenue of the Americas
New York, New York 10036-4003
BY: BRUCE W. BABER, ESQUIRE

GOOGLE, INC.
1600 Amphitheatre Parkway
Mountain View, California 94043
BY: RENNY HWANG, LITIGATION COUNSEL

For Dr. Kearl:

FARELLA BRAUN & MARTEL LLP
235 Montgomery Street, 30th floor
San Francisco, California 94104
BY: JOHN L. COOPER, ESQUIRE

Also Present:

SAFRA CATZ, President and CFO
Oracle Corporate Representative

CATHERINE LACAVERA
Google Corporate Representative

- - -

P R O C E E D I N G S

APRIL 23, 2012

7:29 a.m.

(Proceedings held in open court, outside
the presence and hearing of the jury.)

THE COURT: Be seated.

Thank you for your briefs. I've read them both. Let
me make a couple of comments.

There's still so much there that I'm not sure I'm
ready to say I've mastered everything in those briefs, but one
heads-up I need to make sure you understand; and that is, you
did what I asked for and you referred to various parts of the
record, but if it's going to be something I make a final
decision on in this case, it has to be in the trial record. It
will not do you any good to have lurking as a stipulation
somewhere in the past. The stipulation has got to be read to
the jury.

And, likewise, those points that have been
established by order. Those have to be read to the jury or
they fall away.

So you must know that and get it -- you know, there's
still time for you to get whatever you want by way of
stipulation before the jury and make it part of the trial
record, but what I don't want to be in the position of is at
the time that I have to make a decision and the record is

1 closed for trial, somebody is saying, "Oh, remember that
2 interrogatory?" Or, "Oh, remember that brief where they
3 admitted X?" I just can't run a railroad that way.

4 The whole point of a trial is to put in the proof
5 that you think governs your case. I've said this before. I
6 think you should understand it by now, but I say it again.

7 All right. There is an issue that you, I don't
8 think, have helped me much on, which is judge versus jury on
9 what is the work as a whole? Whether the jury decides that or
10 the judge decides that? I would like for you to do more work
11 on that because I'm working on it. I have some of my team,
12 much smaller team than you have, but I am not convinced that
13 this is for the judge to decide.

14 One way to go is to say Oracle copyrighted 5.0 and if
15 it wants to establish a smaller work as a whole for purposes of
16 whatever, you know, it's a matter of proof. Argue it to the
17 jury.

18 Likewise, same for the defendant. I'm not saying
19 that that's the way I'm going to rule. I'm just saying that
20 I'm turning over in my mind how we deal with the work as a
21 whole issue. You disagree on it. Is that for me to decide and
22 then just instruct the jury, as Oracle says, or is that for the
23 jury to sort out and you, as good lawyers, to argue your
24 respective cases?

25 Here is a question I have for you, and it concerns

1 the 37 APIs. And, again, I ask you these things. You need to
2 be answering in the spirit of "this is what the evidence in the
3 trial will show," not arguing sort of offline something that
4 the jury is never going to hear.

5 I would like to know what the evidence is going to
6 show -- and let me first direct this to Google.

7 It seems to me that Google is admitting that it
8 copied the 37 declarations into its Android, is that part true?

9 **MR. BABER:** Your Honor, I don't want to quibble with
10 the word "copy." We used the --

11 **THE COURT:** You have got to come up here.

12 All right. Put it in your own words, but be as frank
13 and candid as you can.

14 **MR. BABER:** Yes, your Honor.

15 The only reason I quibble with the word "copy" is
16 because these APIs and their specifications have been out for
17 a long time. They are available in lots of places. They are
18 in books. They are on the web, et cetera. They are able at
19 Harmony, at GNU Classpath.

20 But is no question we absolutely used and included in
21 Android the 37 API packages and the specifications so that we
22 would have them exactly right. Exactly so that they complied
23 with the specifications.

24 **THE COURT:** All right. So, here is the -- part of
25 what I don't understand is: What is so special about these 37?

1 Could it have been 34? Could it have been 42? What's so
2 special about these 37? Because you say in other places you
3 did not use all of the then existing APIs, right?

4 **MR. BABER:** That's correct.

5 **THE COURT:** So what's so special about these 37?

6 **MR. BABER:** Your Honor, as of right now, what's so
7 special about these 37 is that they are the subset of the Java
8 APIs in Android as to which Oracle is asserting rights.

9 And let me back up to tell you, there are about 51
10 Java packages in Android. When we started this lawsuit, they
11 accused all 51. But then we pointed out to them, well, wait,
12 there are 14 or 15 in here that you don't own. Somebody else
13 wrote this and they dedicated it to the public before they gave
14 it to Java, or they retained rights to it. So they weren't in
15 the position to even assert rights as to those 14 and they
16 dropped them.

17 So what makes the 37 the 37 at this point, is that
18 they are the ones left out of where they started, where they
19 think they can assert ownership rights. So that's how we get
20 to 37.

21 The reason there are 51 in Android in the first
22 place -- and you'll hear testimony on this from our witnesses
23 during our case -- is they started with all of the Java APIs.
24 There were some, frankly, and you've heard the testimony about
25 the Java Platform and how it was not appropriate for use in a

1 smart phone. So there were lots of packages that would make no
2 sense to put them in Android; the user interface, some of the
3 other things that make a smart phone a smart phone. So there
4 was no reason to put those in Android.

5 Then there were others where we were doing our own
6 Android specific APIs, so there was no need for the Java ones.
7 We had better ones that we wanted to put in for our APIs.

8 So you started with all of them. You took out the
9 ones that weren't applicable. You took out the ones where we
10 had better ones. And what you had left was this collection of
11 51 that were certainly all of the core ones. You have heard
12 about testimony --

13 **THE COURT:** Well, I don't know what "core" means.
14 What does that mean?

15 **MR. BABER:** You've heard testimony, your Honor, I
16 think from had Dr. Reinhold and from Dr. Bloch and others that
17 in the Java language specification it tells you that these four
18 packages were the original four packages that were part of the
19 language from day one. In order to use the Java language, you
20 had to have Java.lang. That's language. Java.util, that's
21 basic utilities. Java.io, input/output.

22 We had testimony from Dr. Reinhold from Oracle and
23 Dr. Bloch from Google that without IO you might could write a
24 simple program, but you couldn't even send to it a printer.
25 You couldn't send to it a disk. That's input/output, very

1 basic to the language.

2 And the fourth core package is Java.net, which lets
3 you get things on the internet, and Java was written to take
4 advantage of the internet. That's how they touted it when it
5 first was written. So, lang, util, IO and net have always been
6 referred to as the core packages.

7 **THE COURT:** That's four?

8 **MR. BABER:** Those four.

9 **THE COURT:** But a moment ago you said 37 were the
10 core.

11 **MR. BABER:** There are 37, your Honor, and among
12 the --

13 **THE COURT:** You mean 37 equals the core or four
14 equals the --

15 **MR. BABER:** Four are the core. Four of the 37 are
16 the star traditional core packages.

17 The other 33 are a combination of things. Some are
18 subpackages under those four. There are packages with names
19 like java.lang.something else, java.util.something else.

20 But the 37, you'll hear testimony from the Google
21 witnesses that they wanted to include in Android all of the
22 basic packages that programmers would expect to have available
23 to them when they were writing programs in the Java language.
24 So that's how we wound up with 51 in Android and then only 37
25 of the 51 accused.

1 **THE COURT:** Right. At the time that this was done,
2 the Android was being written, did the Java language only have
3 51 APIs or were there more?

4 **MR. BABER:** 166, your Honor.

5 **THE COURT:** At that point? Not now, but back then?

6 **MR. BABER:** Well, actually -- Java 5.0 had 166.

7 Now, between 5.0 and the time Android was being
8 worked on, they might have added -- there were even more in
9 Java 6 and Java 7. I don't know how much, if any, it had grown
10 between the five --

11 **THE COURT:** Let's say, in 2008 how many were there on
12 the Java side?

13 **MR. BABER:** There were at least 166.

14 **THE COURT:** So somebody at Google went through that
15 list and figured out which ones a programmer would, quote,
16 expect, close quote, to see in Android? That's what I'm
17 asking. Is that the way you want to put it?

18 **MR. BABER:** Yes, your Honor. But subject to the
19 caveat that first they had to exclude the ones that really
20 wouldn't be any good for Android. All right? Android has a
21 special user interface facility. It makes it a smart phone.
22 It makes it attractive to users.

23 Well, any of the Java APIs that might have worked on
24 feature phones, like you have heard the testimony, might be
25 good on those old little flip phones, that was inadequate for a

1 smart phone. So Android spent a lot of time developing a new
2 user interface.

3 So it would make no sense. Even if programmers
4 expected them, programmers would know that, look, Android is
5 something new. It has a new user interface. In order to
6 access the user interface, I've got to go to the Android APIs.
7 That's not going to be in my Java namespace that's going to be
8 over here in Android because that's one of the things that
9 makes Android such an improvement over the Java Platform.

10 **THE COURT:** All right.

11 **MR. BABER:** So they first took out the ones that
12 wouldn't work and then they went to the ones programmers would
13 expect.

14 **THE COURT:** So at some point around 2008, I'm
15 assuming, Google identified 51, true?

16 **MR. BABER:** Correct.

17 **THE COURT:** And then of the 51, 14 were in the public
18 domain or something?

19 **MR. BABER:** Either in the public -- somewhere
20 expressly in the public domain. There are some -- for example,
21 you heard testimony last week about this Professor Doug Lee in
22 New York, who has been a big player in the Java community. He
23 wrote several -- I think, three of the packages:
24 Java.util.concurrent and java.util.concurrent.atomic,
25 java.util.concurrent.something else.

1 Those three packages were originally accused. We did
2 the research and said, Wait a minute. Doug Lee wrote these and
3 expressly dedicated them to the public and gave them to Java at
4 the same time. Oracle decided, Okay, maybe we won't assert
5 that.

6 **THE COURT:** Then of the 37, you're saying that four
7 were embedded in the basic programming language?

8 **MR. BABER:** At least four, your Honor. You have
9 heard testimony, again, from Dr. Reinhold and Dr. Bloch, and
10 what they both did was they started with the language
11 specification. They said, they are the four core packages for
12 sure, but the Java language book itself identifies a whole lot
13 of other classes, methods, et cetera.

14 So in order to implement the language, you have to go
15 through the book -- at a minimum, you have to go through the
16 book, find out all the ones that are referenced in the book
17 about how to use the language, and those have to be included as
18 well.

19 And both Dr. Bloch and Dr. Reinhold testified that
20 that's on the order of -- I forget the numbers candidly, your
21 Honor, but dozens and dozens of classes spread across eight or
22 so, as I recall, different packages. So if you go through that
23 basic exercise of just starting with the published book on the
24 language, you get to a starting universe of more than just the
25 four. And then Dr. Bloch explained that to do it right, the

1 book tells you about a certain class. So you say, okay, that
2 class has to be in here.

3 **THE COURT:** What book is this?

4 **MR. BABER:** This is the book, your Honor, in
5 evidence, the big -- the *Java Language Specification* book.
6 It's the one that defines the language.

7 **THE COURT:** Okay.

8 **MR. BABER:** So what Dr. Bloch testified is, you go to
9 the book. You find out which classes the book tells you about.
10 Then you go to see how that class is defined, what its
11 specifications are.

12 And in many, many occasions you find that in order to
13 implement that class, you need lots of other things as well.
14 You can't just do it and say, "I'm done." But in order to get
15 it right, you have to go get a bunch of other classes from
16 other packages.

17 And when you do all that analysis and you look at
18 what it takes to fully implement the language, you wind up
19 literally with thousands of these things, thousands of them
20 spread across, I believe, it's 12 of the packages in order to
21 fully and completely implement the language.

22 **THE COURT:** The programming language?

23 **MR. BABER:** The programming language, yes, your
24 Honor.

25 **THE COURT:** Wait a minute. Let me remember my

1 thought.

2 (Brief pause.)

3 **THE COURT:** Okay. Again, I say to you, some of the
4 things you've said this morning I've never heard before. It
5 doesn't count until there's proof. So you need to prove
6 whatever you're going to prove at trial.

7 This is good to educate me because I'm thinking about
8 all these issues, but at the end of the day if you fail to put
9 in some of these points, you know, that's your problem. So be
10 sure you prove whatever you think is necessary for your case.

11 **MR. BABER:** Your Honor, may I just -- I just want to
12 make sure, I want to be as precise as I can be. I want to
13 clarify one little point I said earlier.

14 I was talking, for example, about the Doug Lee
15 packages that he wrote and dedicated to the public. When I say
16 "public domain" or "owned by others," I'm generally talking
17 about that code, the implementation.

18 It's our view, of course, that the API specifications
19 are in the public domain, as all of them. They are not
20 copyrightable. Anyone can use them, whether it's Classpath or
21 Harmony or Doug Lee or Google or IBM or anyone else. It's only
22 when you get to that implementation, the code, that we think
23 there can be any enforceable copyright rights. So that's a
24 legal argument.

25 **THE COURT:** I've already said that I'm going to

1 instruct the jury, subject to a motion under Rule 50 later at
2 the end of the case. I'm going to instruct the jury that the
3 copyrights extend to the Structure, Sequence and Organization.

4 Now, I'm reserving on that ultimately. I see both
5 arguments on that point, but we ought to get the verdict on
6 that. If you were to win on fair use, for example, then the
7 judge doesn't have to decide those hard questions.

8 **MR. BABER:** I understand.

9 **THE COURT:** But, so the way it's going to be
10 presented to the jury is that the Structure, Sequence and
11 Organization is covered by copyright. Not its own copyright.
12 I'm not prepared yet to say to the jury somehow that that is a
13 stand-alone entity that has its own separate copyright, but I
14 will say that the copyright that was issued by the copyright
15 office extends to the Structure, Sequence and Organization,
16 along with all the literary words and so forth.

17 All right. You've given me your side. I want to
18 hear from Mr. Jacobs.

19 Mr. Jacobs, I want you to respond to anything you
20 want, but here is, I think, the tougher question for you. I
21 sometimes get the impression that what you did was you went
22 through and you found the ones that -- where there are -- the
23 37 where they did copy the declarations, and you're trying to
24 convert that into some sort of Holy Grail of copyright law;
25 that it's some kind of stand-alone work when I don't see that

1 yet. The proof doesn't show that yet.

2 So I'm curious as to why you would want the judge to
3 instruct the jury that the work as a whole is these 37, when so
4 far that's not the actual evidence.

5 Go ahead.

6 **MR. JACOBS:** Two different issues lurking there. You
7 said a couple minutes ago, "I'm not going to instruct the jury
8 that Structure, Sequence and Organization is a separate thing."
9 And I think that's absolutely correct. The Structure, Sequence
10 and Organization of the copyrighted work is a copyrightable
11 element of the work or it's the subject -- it's --

12 **THE COURT:** Something like that is the way we're
13 going to put it to the jury and let them go on the assumption
14 that that's correct. I don't -- maybe not by your exact
15 wording, but the concept I agree with.

16 **MR. JACOBS:** Then the second question is the 37
17 packages. And I think the way these cases line up, and we did
18 do a lot of case reading --

19 **THE COURT:** You didn't give me a case on point. You
20 know, the fact that the Ninth Circuit affirms a summary
21 judgment where the issue was not raised, but could have been
22 raised, does not mean that the Ninth Circuit ruled that it was
23 a matter of law for the judge. Nobody raised it in that case,
24 or did they?

25 **MR. JACOBS:** Probably the -- well, I think the case

1 that jumped out for me, your Honor, is the *Texaco*, the *American*
2 *Geophysical* case. That's the one where the -- where Texaco was
3 copying articles from a journal internally. And they teed it
4 up as kind of a test case to see whether if a company copies
5 articles for internal distribution, that is a fair use issue or
6 a -- whether that's subject to fair use.

7 The case is very interesting on fair use grounds
8 because it talks about the fact that the publisher has a
9 license for internal copying. It's interesting on work as a
10 whole grounds because the registered entity was the entirety of
11 the journal. What Texaco was copying internally were
12 particular articles, and it was the particular articles that
13 were the focus then of all of the analysis, but in most
14 particularly fair use.

15 There was no credible argument -- it was raised, but
16 not adopted by the Court that, look, all they are copying at
17 Texaco are the individual articles that they are most
18 interested in because they are the most interesting to their
19 scientists and circulating those internally, akin to the most
20 interest to the developer's argument that you just heard from
21 Mr. Baber.

22 The Court says no, no, no. We don't -- we --
23 although this wording isn't used, I think the spirit of the
24 decision is we take the defendants -- the defendant, in effect,
25 defines the work by that which it chooses to copy. We're not

1 going to let the defendant off the hook because the defendant
2 didn't copy all of a registered work. If the defendant thought
3 it was meaningful to copy the article from the journal, then
4 that's the work we're going to look at.

5 **THE COURT:** Well, then, that will always be
6 100 percent. If you can, you know, cut and paste like that and
7 use a pair of scissors and just cut out the part that was
8 copied, then the comparison to the work as a whole is always
9 going to be one to one, 100 percent.

10 **MR. JACOBS:** And if you look at the *Nation* case
11 involving Gerald Ford's memoirs, you get pretty close to that
12 conclusion. They copied a couple hundred words when they
13 pre-published in the magazine Gerald Ford's memoirs -- yes, his
14 memoirs. And the Court said, you know, we better look at that
15 body of material and look at whether it's substantial.

16 Now, there are little leavening principles here. I
17 don't want to take this to an extreme, but in the case at hand
18 we know how these packages are developed. They start with a
19 JSR. They are a body. The package is a body of work. It's
20 not an artificial creation. And then when you look at the 37
21 packages, well, those are the packages that Google decided to
22 take that they are akin to 37 journal articles.

23 So I don't think we're stretching it very far to say,
24 look at the packages that they took and make one's analysis
25 based on that. And I don't see any other way to meaningfully

1 instruct the jury on the point that a plagiarist does not
2 escape by adding or -- adding material.

3 **THE COURT:** We can say that they don't escape, but
4 the statute calls out the comparison of the work as a whole for
5 fair use, right? I mean, that's in the statute. I've got it
6 right here. I can read it to you. This 107 says that one of
7 the factors our Congress said must specifically be considered:

8 "The amount and substantiality of the portion
9 used in relation to the copyrighted work as a
10 whole."

11 So you would read that out of the statute because in
12 your analysis you only look at and make a comparison against
13 what was copied.

14 **MR. JACOBS:** In the case of dead literal copying,
15 that provision does not get very much weight in the case law,
16 your Honor. Where you take five notes out of a song, even
17 though it's a large song, you don't get a fair use -- fair use
18 doesn't take you very far, even though it's five notes out of a
19 song. And that's what the decision allowed in --

20 **THE COURT:** That was *Leonard Hand* in 1927. This is a
21 statute that came into effect in 1976.

22 **MR. JACOBS:** But we have lots of snippet cases. We
23 will do some more briefing, your Honor, on this.

24 The other possibility is to say to the jury: The
25 work as a whole is this, but don't consider that which Google

1 added when it took the 37 packages. Just consider the 37
2 packages. That strikes us as pretty confusing.

3 **THE COURT:** We ignore, I think, possibly what -- the
4 add-ons that Google put in, but the part that they copied, the
5 comparison is to your work as a whole, whatever that is. And I
6 question whether the work as a whole is just what was copied
7 from that work as a whole.

8 **MR. JACOBS:** I think we'll do our best, your Honor,
9 to -- maybe we should get a briefing schedule from you.

10 **THE COURT:** Well, I think you should -- both sides, I
11 would like to -- you know, I'm working on it now. I'm working
12 on -- I need your help and you need to give me a decision on
13 point that says: Is this for the judge, or is this for the
14 jury?

15 One way to go is for you excellent lawyers to argue
16 to the jury what is the work as a whole, and each side will
17 have their argument on that. Then I say to the jury, consider
18 that factor.

19 And that's -- you know, I think that's the way most
20 judges, in fact, would go. But I'm going to follow whatever
21 the law is here, so give me decisions on point that say the
22 judge must make that determination and instruct the jury as to
23 what the work as a whole is. Okay?

24 **MR. JACOBS:** We'll do our best, your Honor.

25 **THE COURT:** All right.

1 **MR. JACOBS:** But we do have a technical disagreement
2 about the language and the classes, but I don't want to get
3 into that now. We don't have the time.

4 **THE COURT:** We have a few minutes to go over any
5 issues you want to do before we bring the jury back in.

6 **MR. JACOBS:** Yes, your Honor.

7 We would like to move into evidence some exhibits
8 that the parties have stipulated to.

9 **THE COURT:** All right. Let me get my sheet.

10 Okay. Go ahead.

11 **MR. JACOBS:** TX-26, 104, 157, 233, 234, 251, 21, 172,
12 1029, 238.

13 **THE COURT:** I didn't catch the last one.

14 **MR. JACOBS:** Sorry, it goes backwards. 238.

15 **THE COURT:** 238?

16 **MR. JACOBS:** And then a few more --

17 **MR. VAN NEST:** No objection on those, your Honor.

18 **THE COURT:** All received.

19 (Trial Exhibits 26, 104, 157, 233, 234, 251, 21, 172,
20 1029 and received in evidence)

21 **THE COURT:** Okay.

22 **MR. JACOBS:** TX-34, 165, 383, 384, 416.

23 **MR. VAN NEST:** No objection, your Honor.

24 **THE COURT:** All received.
25

1 (Trial Exhibits 34, 165, 383, 384 and 416 received
2 in evidence)

3 **MR. JACOBS:** 897.

4 **MR. VAN NEST:** No objection.

5 **THE COURT:** Received.

6 (Trial Exhibit 897 received
7 in evidence)

8 **MR. JACOBS:** 729, and then a whole bunch of 896
9 points. So 896.1, .2, .3, .4, .5, .6, .7, .8.

10 **MR. VAN NEST:** No objection.

11 **THE COURT:** All received.

12 (Trial Exhibits 729, 896.1, 896.2, 896.3, 896.4,
13 896.5, 896.6, 896.7, 896.8 received in evidence)

14 **MR. JACOBS:** We will get you our brief comments on
15 the findings of fact and conclusions of law today, but we agree
16 with Google on Google's timing comment about your procedure for
17 findings of fact and conclusions of law.

18 **THE COURT:** Did I get that yet? Did you submit
19 something on that?

20 **MS. ANDERSON:** Yes, your Honor. We did submit it
21 last night.

22 The only comment we had on your Honor's order was we
23 request that the first briefing be due the second business day
24 following close of evidence.

25 **THE COURT:** I'll consider it, but I may have to make

1 some rulings pretty quickly and if they turn on yours, then we
2 have a hiatus. So I'll consider that.

3 **MS. ANDERSON:** Thank you.

4 **MR. JACOBS:** I think we could do the other items at
5 the next break, with the exception of one item that Mr. Fred
6 Norton has and an item that Mr. Boies has.

7 **MR. VAN NEST:** Your Honor, there actually were three
8 more exhibits, I thought, that we had stipulated coming in.

9 **MR. JACOBS:** They are your exhibits though.

10 **MR. VAN NEST:** I thought we would just do them all at
11 ones.

12 **THE COURT:** Go ahead.

13 **MR. VAN NEST:** TX-612, TX-617, TX-2223.

14 **MR. JACOBS:** No objection.

15 **THE COURT:** All received in evidence.

16 (Trial Exhibits 612, 617 and 2223 received
17 in evidence)

18 **THE COURT:** All right. More issues?

19 **MR. BABER:** Yes, your Honor.

20 **THE COURT:** We have got two minutes, so give me the
21 ones that are important and we're going to move on with the
22 jury. We can't have all these miscellaneous things.

23 Okay, Mr. Norton.

24 **MR. NORTON:** Thank you, your Honor.

25 On Friday we moved the admission of Exhibit 1026, the

1 Danger license. The Court said it would hold that in abeyance
2 until we had an opportunity to discuss it outside the presence
3 of the jury.

4 We filed a motion last night, a brief in support of
5 the admission of 1026 and Google has opposed. We would like to
6 get that exhibit in for the reasons we've stated. I'm prepared
7 to address that now.

8 **THE COURT:** Go ahead.

9 **MR. NORTON:** This is a license that Mr. Rubin entered
10 into while he was at Danger with Sun, and the reason why he
11 entered into it was that Sun told him he must take a license
12 even though he said -- the testimony from Mr. Swetland is
13 unchallenged -- Danger had never touched Sun's source code.
14 They had not taken any source code at all. And, yet, they
15 entered into what is effectively a specification license and
16 they customized the document, Exhibit 1026. They made it into
17 a specification license.

18 And that document is relevant to make it clear to the
19 jury -- this was your Honor's question on Friday afternoon:
20 When Google says we need a license, are they talking about a
21 license to source code, or a license to specification
22 specifications?

23 And it's quite clear from this document, as well as
24 other evidence, that Mr. Rubin was quite aware that it was
25 Sun's position that you needed a license exclusively for the

1 specifications. In fact, in 2003 he caused Danger to enter
2 into such a license based on nothing, but Danger's use of
3 specifications.

4 And Google doesn't want that exhibit to get in front
5 of the jury, even though that's precisely what it was. So it's
6 relevant. It's relevant to issues in Phase 1.

7 **THE COURT:** So you're saying it's relevant to show
8 that Mr. Rubin, who later went to Google and had something to
9 do with Android, was on notice of what Sun's position was as to
10 that you need a license for specification; is that your point?

11 **MR. NORTON:** It is relevant to his knowledge. It is
12 relevant that he, in fact, had taken such a license, which
13 establishes -- you know, Mister --

14 **THE COURT:** Why is that different from what I just
15 said?

16 **MR. NORTON:** Well, it is --

17 **THE COURT:** See, when a lawyer won't agree to the way
18 I phrased it --

19 **MR. BOIES:** Just say yes.

20 **THE COURT:** (Continuing) -- then I just feel like
21 your splitting hairs and I don't understand the point.

22 **MR. NORTON:** I agree with your Honor.

23 And I want to make an additional point. Mr. Van Nest
24 argued to the Court on Friday that this whole idea that Sun's
25 position was that APIs were copyrightable is something that

1 Oracle invented for purposes of this lawsuit. That was his
2 statement to the Court on Friday afternoon. And that's simply
3 not true.

4 Not only was it Sun's position, it was a position
5 that was acknowledged by Danger when Mr. Rubin caused Danger to
6 enter into a license and pay money because Danger had been
7 using Sun's specifications. And then that knowledge is
8 relevant for the reasons we stated.

9 **THE COURT:** So why isn't that correct?

10 **MS. ANDERSON:** Thank you, your Honor. Christa
11 Anderson.

12 Your Honor, the question is not whether certain
13 questions can be asked of witnesses about Danger taking a
14 license and why. Obviously, your Honor has heard testimony on
15 that from multiple witnesses, including Mr. Cizek, that Oracle
16 called.

17 The problem is whether Exhibit 1026 comes in
18 evidence. This is a multi-page, single-spaced extremely
19 complicated license agreement for a completely separate company
20 about a completely separate platform. There are provisions in
21 that agreement for different rights that were supposedly
22 conferred on Danger under that agreement, including rights to
23 use the Java trademark. So this agreement, if it comes in
24 evidence before the jury, presents a significant risk of
25 confusion.

1 The Danger platform is not what is at issue in this
2 lawsuit. And Oracle's counsel, putting this document before
3 the jury to invite them to try to speculate what these very
4 dense legal provisions in this agreement mean with respect to a
5 platform that nobody is examining as part of this case, really
6 presents a 403 problem.

7 **THE COURT:** But the point that Mr. Norton made seems
8 like a decent point; and that is, that Mr. Rubin was on notice
9 that Sun required a license to even use the specification.

10 **MS. ANDERSON:** Actually, your Honor, Exhibit 1026
11 doesn't show that. Exhibit 1026 shows that what that license
12 covered was, among other things, rights to use a trademark and
13 rights to receive code for what they call a shared part.

14 **THE COURT:** It doesn't use the word "specification"?

15 **MS. ANDERSON:** The point that is cited -- the
16 document that, my understanding, Oracle cites for that
17 proposition is a completely separate SEC filing. It's not the
18 license.

19 **THE COURT:** Wait. Mr. Norton, is that true? Show me
20 the word "specification" in the license.

21 **MR. NORTON:** The word "specification" does not appear
22 in the license, but what the license does in Attachment F,
23 which Mr. Cizek said was specifically requested by Mr. Rubin,
24 it says they do not get rights to source code.

25 But what is -- the significance of the license is not

1 merely what it provides for going forward, but the fact that
2 Mr. Swetland testified that prior to -- and Mr. Cizek
3 testified, that before Danger entered into the license, it had
4 done its own independent implementation and then Sun, and
5 Mr. Lindholm, approached Danger and said, "You cannot do this.
6 Even though you have not seen our source code, you must take a
7 license. Even though you have not used our brand, you must
8 take a license."

9 So an independent implementation by Danger required a
10 license, and the license that was entered into is Exhibit 1026.

11 **THE COURT:** But if it doesn't -- if it doesn't use
12 the word "specification" -- you told me it used -- you didn't
13 tell me it used the word "specification," but you said he took
14 a license for the specification.

15 Now I'd like to know -- Mr. Boies is coming forward
16 to rescue you and, apparently, he has found the word
17 "specification."

18 What did he find?

19 **MR. NORTON:** That's exactly right. Mr. Boies has
20 rescued me.

21 On Page 10 of Exhibit 1026 it specifically calls out
22 the original code, the upgraded code and specifications.

23 **THE COURT:** All right. Ms. Anderson, why did you
24 tell me the word "specification" was not in there?

25 **MS. ANDERSON:** Actually, your Honor, my understanding

1 was it was not in. In fact, the document that Oracle cited for
2 this proposition was an SEC filing. It was not the document
3 itself.

4 **THE COURT:** Why did you tell me it wasn't in the
5 license agreement?

6 **MS. ANDERSON:** That was my understanding, your Honor,
7 that --

8 **THE COURT:** You're wrong, it looks like.

9 **MS. ANDERSON:** Yes. However, it does not change the
10 fundamental point, your Honor.

11 The point is this jury shown this document with many,
12 many pages of complicated provisions regarding legal issues
13 will be invited to speculate and try to apply these terms to
14 this particular platform when it is not the same platform. It
15 was an agreement that was done in 2003.

16 There is nothing about this agreement that will
17 change the fact that Danger took a license to have rights to a
18 trademark and did receive code, which is contradicted by
19 what -- which is contradictory to what Mr. Norton just told
20 you. They received code for a shared part, and there is
21 testimony about it, and it's in the license agreement. So what
22 this is going to become is a rabbit hole of confusion.

23 Now, that doesn't mean that Oracle's counsel can't
24 ask questions of witnesses, which they have already done, and
25 I'm sure they intend to ask Mr. Rubin what was his

1 understanding and negotiations related to the Danger license.
2 That has its own potential for confusion.

3 But, certainly, with this license agreement in
4 evidence, there will be significant risk of confusion,
5 particularly where we have already complex issues that are
6 presented to the jury for decision.

7 **THE COURT:** Mr. Van Nest, are you --

8 **MR. VAN NEST:** I was going to make a suggestion in
9 light of the hour, your Honor, that Mr. Rubin should be on
10 today. Why don't the Court wait and see what Mr. Rubin says
11 about it and you can make your judgment?

12 **THE COURT:** The foundation has already been laid for
13 1026 through that other witness, correct?

14 **MR. NORTON:** Yes, it has, your Honor.

15 **THE COURT:** All right. Here is the answer. 1026 is
16 allowed in evidence. Mr. Boies saved the day. I was going to
17 exclude it until Mr. Boies pointed out the word "specification"
18 is in there, contrary to what Ms. Anderson had said.

19 Ms. Anderson, you shouldn't have done that. You
20 should know the document by heart. I know you're working hard
21 on this case and I respect you and all of that, but I have to
22 rely on the lawyers to pitch their arguments in accordance with
23 the record. So, please, all of you, do your best to be as
24 accurate as you can.

25 It's time now to bring the jury in and resume.

1 (Trial Exhibit 1026 received
2 in evidence)

3 **MR. BABER:** Your Honor, we have an issue that relates
4 to the testimony the very first witness will give, I believe
5 the very first part of his testimony.

6 **THE COURT:** What is that?

7 **MR. BABER:** Your Honor you will recall on Friday Bob
8 Lee took the stand and we adjourned his examination because of
9 the exhibit we had been given just late the night before.

10 **THE COURT:** Yes.

11 **MR. BABER:** Your Honor, we have an objection, a
12 significant objection to that exhibit. We certainly don't
13 object to Mr. Jacobs questioning the witness about the
14 documentation that is in Android. You heard testimony he was
15 the core library lead for some time.

16 We do object to him doing it in front of the jury in
17 a manner that the Ninth Circuit has said is misleadingly
18 prejudicial. I will explain to the Court.

19 If I can hand up, this is what we understand they
20 want to use with this witness.

21 (Whereupon, document was tendered
22 to the Court.)

23 **MR. BABER:** That's a printed copy of what, I assume,
24 will be a series one-by-one of side-by-side comparisons of a
25 little snippet from Java and a little snippet from Android.

1 **THE COURT:** Wait a minute. Is this a -- is this
2 something gemmed up just for the litigation?

3 **MR. BABER:** Absolutely, your Honor. And scissors
4 were --

5 **THE COURT:** Not on a fact witness. No. You're going
6 to use the original documents. This is an argument piece.

7 I'm not going to allow this to be used with a
8 witness. Not on a fact witness. Your experts can do it maybe,
9 but not a fact witness.

10 **MR. BABER:** Thank you, your Honor.

11 **THE COURT:** Now we have a note from the jury, one
12 juror.

13 Is the jury ready to come in?

14 **THE CLERK:** Yes.

15 **THE COURT:** Would you line them up, please? Are they
16 ready?

17 **THE CLERK:** They are ready.

18 **THE COURT:** Here is the note from a juror unsigned.
19 Don't bring them in yet.

20 "What if one of us juror people" --

21 No.

22 "What if one of us jury people were to be hit
23 by a truck - or something less dramatic, but
24 equally capable of keeping us from being
25 here? Maybe what I'm really asking is this:

1 Why aren't there alternates for this trial?"

2 A very good question, and I will answer it when they
3 come in.

4 All right. Let's bring in the jury.

5 (Jury enters courtroom at 8:08 a.m.)

6 **THE COURT:** Okay. Be seated, please.

7 So, good morning. Did you all have a good weekend?

8 (Jury nodding affirmatively.)

9 **THE COURT:** Everyone washing your hands like crazy?
10 Staying in good health? Yes?

11 (Jury nodding affirmatively.)

12 **THE COURT:** I don't want anyone to get sick.

13 Now, one of you in that connection wrote me a note.

14 I don't need to know who it is, but it asks why is it we don't
15 have alternate jurors on this jury?

16 Well, that's a very good question. In criminal cases
17 we do have alternate jurors. In civil cases, I believe it was
18 a 1990 or '91 -- I can't remember exactly -- Congress did what
19 I think was a wise thing and amended the law to say that there
20 is a minimum of jurors that we need to decide a civil case.

21 This, of course, is a civil case, not a criminal
22 case. And so we start with a few more than we need by the
23 statutory minimum. So if one of you were to -- I won't use the
24 example you gave here, but if one of you were to get very ill
25 and have to drop out of the trial, we could go forward with 11.

1 And we could actually go forward with a number less than that,
2 but I don't want to start speculating -- getting you to
3 speculate on anything. So I don't want you to -- I'm just
4 going to -- it's a secret. It's a trade secret what the number
5 is.

6 (Laughter.)

7 **THE COURT:** But I know that none of you are going to
8 get sick because you're being good and healthy and none of you
9 are going to have an accident, we hope. Though, you know, some
10 of you are coming a long way every morning and I do worry on
11 these cases a little bit that somebody may on a dark morning
12 coming to serve their country by serving as a juror in this
13 case or any other case, they might get in an automobile
14 accident, and I do worry about that.

15 So I don't take your -- none of us here take your
16 participation lightly here. It is a burden on you to serve.
17 There is no question about it.

18 However, if one of you did have to drop out for
19 whatever reason, we could go forward with 11. We could go
20 forward with 10. I'm not going to say exactly where the cutoff
21 line is, but the Congress of the United States has answered
22 that question for us and did a smart thing by allowing us to
23 start with a larger number and then allow for flexibility if
24 somebody had to drop out. Okay?

25 Good. So we're back to work this morning, this nice

1 pretty Monday morning. This week will be -- I think we're
2 going to come close to finishing the evidence on Phase 1. We
3 may not quite finish it, but we'll get close. And I'm positive
4 that next week the case will go to you -- the first phase will
5 go to you for decision. So we're on track.

6 Now, you remember we had a witness last week whose
7 name was -- I'm embarrassed...

8 **MR. JACOBS:** Mr. Lee, your Honor.

9 **THE COURT:** Mr. Lee. Yes, Mr. Bob Lee. And we had
10 to interrupt him because at the last minute the Oracle side had
11 discovered some documents they wanted to use with him, and I
12 have a procedure that there's some minimum notice needs to be
13 provided to the other side. Nobody did anything wrong here.
14 It's just one of those things that happened in trials. So they
15 both agreed that he would be postponed until today. And now
16 everyone has had a chance to look at the documents and to be
17 ready to do their cross-examination so it goes smoother in
18 front of you. It takes up less of your time. So that's the
19 whole idea.

20 So, Mr. Bob Lee, is he here? He needs to come back
21 in and to resume his testimony.

22 (Mr. Lee enters the courtroom.)

23 **THE COURT:** Are you Mr. Lee?

24 **THE WITNESS:** Yes, sir.

25 **THE COURT:** You have to forgive me. I don't remember

1 for sure. You were here last week.

2 **THE WITNESS:** Yes, sir, on Friday.

3 **THE COURT:** Okay, great. Have a seat.

4 **THE WITNESS:** Thank you.

5 **THE COURT:** Just let me ask you a couple of
6 questions. I've forgotten.

7 Who do you work for now?

8 **THE WITNESS:** I am the CTO of Square. It's a mobile
9 payments company.

10 **THE COURT:** And did you work at one point for Oracle
11 or Google?

12 **THE WITNESS:** For Google.

13 **THE COURT:** You did, all right. And what did you do
14 there?

15 **THE WITNESS:** I started in 2004. I worked on the
16 Android team for two years and I worked on the core, I guess,
17 Java libraries and that sort of thing. And then I was the core
18 library lead for Android.

19 **THE COURT:** All right. With that background, my
20 notes say that we had 17 minutes of the witness's prior
21 testimony.

22 And, Mr. Jacobs, you may pick it up right there.
23 Thank you.

24 **MR. JACOBS:** Thank you very much, your Honor.
25

1 **BOB LEE,**
2 called as a witness for the Plaintiff herein, having been
3 previously sworn, resumed the stand and testified further as
4 follows:

5 **DIRECT EXAMINATION RESUMED**

6 **BY MR. JACOBS:**

7 **Q.** Good morning, Mr. Lee.

8 **A.** Good morning.

9 **Q.** We were talking at the end of the your testimony the other
10 day about the instructions you gave to your documentation
11 writers or coders about what they should do with the narrative
12 portion of the Java specifications; do you recall our
13 back-and-forth on that?

14 **A.** Yes, I do.

15 **Q.** And we were talking about the two possible meanings of
16 paraphrasing; do you recall that?

17 **A.** I don't recall the two meanings. I think I said that it's
18 taking something and putting it into your own words.

19 **Q.** Okay. Let's go with that interpretation as we go through
20 this.

21 I'm going to ask you to take a look on your screen.
22 On the left side you will be seeing Trial Exhibit 610.2, which
23 is the Android documentation -- which is the Java documentation
24 from the Java website; and on the right we're going to be
25 asking you to look at Trial Exhibit 767, which is the Android

1 documentation from the Android website.

2 Do you see that on your screen, sir?

3 A. Yes. These are called JavaDocs and they are generated
4 from the code.

5 Q. And that's true on the Android side as well, correct?

6 A. On the Android side it's called something else. I don't
7 remember what the name of the tool is.

8 Q. And my question wasn't precise. The jury heard an
9 explanation that in Java the way the documentation is created
10 is by a tool that actually reads portions of the source code
11 and then places it in a kind of template that's available on
12 the web as a source of documentation.

13 And so my question is: Is that same mechanism
14 applied for the Android documentation?

15 A. Yes.

16 Q. Now, what we have up here on the screen is the class
17 CipherInputStream; do you see is that?

18 A. Yes.

19 Q. And then we have the javax.crypto.CipherInputStream
20 underneath that; do you see that?

21 A. The -- which one am I supposed to look at?

22 Q. Sorry. Do you see the documentation underneath that,
23 underneath the class CipherInputStream declaration?

24 A. Yes.

25 Q. And so the Java documentation reads:

1 "A CipherInputStream is composed of an
2 InputStream and a Cipher so that read ()
3 methods return data that are read in from the
4 underlying InputStream, but have been
5 additionally processed by the Cipher. The
6 Cipher must be fully initialized before being
7 used by CipherInputStream."

8 Do you see that?

9 **A.** Yes.

10 **Q.** And then if we go over to the Android text for the
11 corresponding Java class in Android, we see the following text:

12 "This class wraps an InputStream and a Cipher
13 so that read () methods return data that are
14 read from the underlying InputStream and
15 processed by the Cipher. The Cipher must be
16 initialized for the requested operation
17 before being used by a CipherInputStream."

18 Do you see that, sir?

19 **A.** Yes.

20 **Q.** Does that documentation in Android meet your definition of
21 taking something and putting it in your own words?

22 **A.** They do look a bit different, yes.

23 **Q.** In what way are they different, sir?

24 **A.** I mean, they are not exactly the same.

25 (Laughter.)

1 Q. They are very similar, aren't they, sir?

2 A. They contain a lot of the same words, that's for sure.

3 Q. Then if we looked at javax.crypto.Cipher and do a similar
4 analysis using the Oracle online documentation on the left and
5 the Android online documentation on the right?

6 THE COURT: These underlying documents are both in
7 evidence?

8 MR. JACOBS: Yes, your Honor. These are 610.2 and
9 767.

10 THE COURT: All right. So the jury will understand
11 that -- do you see that on the screen over there?

12 (Jury nodding affirmatively.)

13 THE COURT: That these two documents that are on the
14 screen side-by-side are in evidence. And somehow through the
15 magic of computers, they are able to pull out parts and compare
16 them side-by-side. That's fine. I think we ought to do that,
17 but remember it's the underlying evidence that counts.

18 And those are in evidence. So you'll have that in
19 mind.

20 Please continue.

21 BY MR. JACOBS:

22 Q. And this text begins:

23 "This class provides the functionality of a
24 cryptographic Cipher for encryption and
25 decryption."

1 Do you see that?

2 A. Yes.

3 Q. That's in the Java documentation.

4 In the Android documentation it reads:

5 "This class provides access to

6 implementations of cryptographic Ciphers for

7 encryption and decryption."

8 Do you see that?

9 A. Yes. They are both concise descriptions of what this
10 class does.

11 Q. And then it goes on to say:

12 "In order to create a Cipher object, the

13 application calls the Cipher's get-instance

14 method, and passes the name of the requested

15 transformation to it."

16 That's on the Java site; correct, sir?

17 A. Yes.

18 Q. And on the Android site it says:

19 "Cipher classes cannot be instantiated

20 directly. One has to call the Cipher's

21 get-instance method with the name of a

22 requested transformation optionally with a

23 provider."

24 Do you see that?

25 A. Yes.

1 Q. And there are individual word differences in there;
2 correct, sir?

3 A. Yes.

4 Q. But they are quite similar in text; aren't they, sir?

5 A. Which sentence? The first sentence or both sentences?

6 Q. Both sentences that are highlighted on the screen.

7 A. I have to count the words. Can you tell me how many words
8 are the same?

9 Q. I didn't count.

10 A. "This class provides the functionality of a cryptographic
11 Cipher for encryption and decryption."

12 This class provides access" -- yep -- "to..."

13 So they both say, "This class provides" -- it's a
14 common thing to say" -- "...functionality of a cryptographic
15 Cipher" to "...implementations of cryptographic Ciphers for
16 encryption and decryption."

17 Yes, that first sentence is very similar.

18 Q. And the second sentence?

19 A. "The application calls for the Cipher's get-instance
20 method." I'm reading it out loud so you don't have to write it
21 all down again.

22 The second sentence looks less similar to me. It has
23 a pretty different structure. Let's see here. "...requested
24 transformation." "...optionally with a provider."

25 The second sentence looks more different to me.

1 Q. Let's take a look at java.io.Channels.pipe.

2 Now, just to step back and give the jury some
3 context. There are thousands of pages of documentation in the
4 documentation for the 37 panels; correct, sir?

5 A. Yes.

6 Q. And that would be true on both the Java and Android side;
7 correct, sir?

8 A. Correct.

9 Q. And the structure of the documentation is identical;
10 correct, sir? And if you think of it as an outline, the
11 outline would match identically; correct, sir?

12 A. Yes.

13 Q. And that's because on the Android side you're documenting
14 the same application programming interfaces as were documented
15 on the Java side; correct, sir?

16 A. Yes.

17 Q. So now if we look at java.nio.Channels.pipe, on the Java
18 said it says:

19 "A pair of channels that implements a
20 uni-directional pipe."

21 And on the Android sides it says:

22 "A pipe contains two channels forming a
23 uni-directional pipe."

24 Do you see that?

25 A. Yes.

1 Q. And then on the Java side it says:

2 "A pipe consists of a pair of channels, a
3 writable sink channel and a readable source
4 channel."

5 And on the Android side:

6 "One is the writable sink channel and the
7 other is the readable source channel."

8 Do you see that?

9 A. Yes.

10 Q. Highly similar? Not so similar? What's your judgment?

11 A. In this case they contain the same words, certainly, but
12 that's to be expected when you're trying to -- you're
13 describing various specific concepts in as few words as
14 possible. You're trying to provide, like, a very concise
15 explanation.

16 Like, for example, the first sentence. What is a
17 pipe, in one sentence. Certainly, if you're trying to do it in
18 that few words, they are going to contain similar words because
19 these are kind of, I guess, the common language or currency of
20 these APIs and simply the technology.

21 Like, "pipe" isn't even actually a -- it's not
22 necessarily a Java term. It predates Java.

23 Q. So you would --

24 A. Like, it's a common technical term.

25 Q. So you would expect the same level of similarity that the

1 jury has seen in these examples so far across the documentation
2 for the 37 packages, sir?

3 A. Generally, yes.

4 Q. Thank you, Mr. Lee. No further questions.

5 THE COURT: Okay. Any cross?

6 CROSS EXAMINATION

7 BY MR. BABER:

8 Q. Good morning, Mr. Lee.

9 A. Good morning.

10 Q. I want to go back just a little bit to some of the
11 testimony you gave last Friday morning just to make sure the
12 jury understands some of the testimony that you gave.

13 In your testimony in response to Mr. Jacobs'
14 questions, you both referred to the phrase, you called it the
15 Java namespace. And the Javax namespace.

16 Could you explain, please, what is a namespace?

17 A. Okay. So I guess at a high level -- the Java programming
18 language in particular, but this is common to a lot of
19 programming languages -- you have types. Those are different
20 types things. Like, you could have something, a car and it
21 might represent a car, or you could have a window and it might
22 represent a window.

23 Then you also have what you call methods. And these
24 are the actual -- the types would be the nouns and then the
25 methods might be like the verbs. So the methods contain

1 implementations that can perform actions. So, like, a window
2 you might have an open method to open that window and a close
3 method to close that window.

4 The way that this stuff works is if somebody else
5 wants to use your window, then they refer to that type by name.
6 So it might be "window." And if they want to use the method,
7 then they refer to that method by name, so they might call it
8 "open." You couldn't call that something else, because you
9 know it's -- the person trying to use it wouldn't be able to
10 use it any more. That name has to be very specific because
11 it's a computer processing this and it has to look up the term
12 exactly by name.

13 When we talk about a namespace, we're just talking
14 about kind of the -- I guess, kind of like a collection of
15 items. In Java this is called a package. Like, you can have a
16 bunch of types in a package and then those packages have names.
17 And typically those -- and for Java, kind of the core packages
18 are -- in the standard packages start with Java and Javax as
19 the name.

20 **Q.** So, for example, when the jury has heard testimony about
21 the java.lang package or the java.util package, does that mean
22 that somewhere in the computer in order to find that function,
23 a programmer has to go access a file that has as part of its
24 name java.util.whatever-method-he's-look-for.

25 **A.** Exactly.

1 Q. Now, in your testimony on Friday you also told Mr. Jacobs
2 you were familiar with TCKs; do you recall that?

3 A. Yes.

4 Q. And just to refresh the jury, what is a TCK?

5 A. TCK stands for Test Compatibility Kit.

6 I guess to take a step back, in the Java world there
7 is something called a Java Community Process. This is a
8 standards organization that comes up with these specs,
9 including the core library spec. It's not just Sun and now
10 Oracle. They don't just create them and throw them over a
11 wall. There's dozens of companies, including Google and
12 myself, that all helped design and contribute to these specs.

13 Part of that process is you -- so if somebody wants
14 to implement one of these specs, there's hundreds of these
15 specs. If somebody wants to implement it and claim to be
16 compatible, they'd have to pass the specific set of tests.
17 These are typically automated that go with that spec. And
18 that's called the Test Compatibility Kit.

19 Q. Mr. Lee, do you know whether it's called a Test
20 Compatibility Kit or a Technology Compatibility Kit?

21 A. Oh, it could be the later.

22 Q. You don't know for sure?

23 A. I don't know.

24 Q. You just called it a TCK?

25 A. We call it a TCK.

1 Q. And at the time Android was developed, was there a Java
2 Platform for a smart phone for which there existed a TCK?

3 A. No.

4 Q. And when you gave testimony to Mr. Jacobs about wanting to
5 be able to run against a TCK, would it have been possible to
6 run Android against a TCK in 2006, 2007 or 2008?

7 A. No. I'm a prolific contributor to these kind of --
8 these -- I guess, these specifications.

9 For example, I created one called
10 JavaDependencyInjection. It was actually the fastest executed
11 specification in the history of the JCP.

12 So it was my intention, being involved with the
13 Android company and heavily involved in the Java community, to
14 create and define this new platform that might support more
15 sophisticated smart phones like this and could possibly be
16 implemented by Apple with the iPhone and that sort of stuff.

17 Q. All right. Could I get -- do you have Exhibit 405 in
18 front of you, Mr. Lee?

19 A. I do not.

20 MR. BABER: May I approach, your Honor?

21 THE COURT: Yes.

22 (Whereupon, document was tendered
23 to the witness.)

24 BY MR. BABER:

25 Q. Mr. Lee, I'm putting back in front you have Exhibit 405,

1 which was admitted, I believe, on Friday and which Mr. Jacobs
2 asked you questions about. Do you recall that?

3 A. Yes.

4 (Document displayed)

5 Q. And just to refresh the jury, what is this document?

6 A. This is an email that I sent to Eric Schmidt, sending him
7 a couple links, telling him about an award I won for an Open
8 Source Java trademark that I created and then asking him about
9 Java and the TCK.

10 Q. So that's the bottom portion of this exhibit you're
11 referring, the part that shows "From: Bob Lee" "To: Eric
12 Schmidt"?

13 A. Yes, sir.

14 Q. I want to direct your attention to the third paragraph of
15 that message you wrote. The second sentence reads:

16 "I'm certain you're already aware of this,
17 but Sun has been abusing their special
18 position within the JCP and playing licensing
19 games with Java SE in order to keep it off
20 phones and protect their Java ME licensing
21 revenues."

22 Did you write that?

23 A. Yes, I did.

24 Q. And what was that -- first let me ask you: That sentence,
25 was that your view at the time?

1 A. It still is, yes.

2 Q. Do you know whether or not anyone else shared that view?

3 A. Everyone, including Oracle, everyone else in the JCP
4 besides Sun.

5 Q. And how do you know that Oracle shared that view, that Sun
6 had been abusing its special position within the JCP and
7 playing licensing games in order to protect its revenues?

8 A. In the Executive Committee -- that's kind of the board for
9 the Java Community Process, this standards organization --
10 Oracle submitted a statement. I don't recall the exact words,
11 but it was asking Sun to stop doing this sort of thing. And
12 then everyone else voted "yes" on it besides Sun.

13 Q. And Google agreed with Oracle in that position?

14 A. Yes. They were allies in that.

15 Q. And let me back up, Mr. Lee, just to -- I don't know if
16 you said this or not, but did you have a formal role with
17 respect to the Executive Committee of the JCP?

18 A. Yes. I was the alternate representative for Google.

19 Q. And who was the actual representative for Google?

20 A. Josh Bloch.

21 Q. Okay. Now, in connection with your involvement -- well,
22 let me back up and stick with the Exhibit 405.

23 You see the part at the bottom -- or the middle of
24 that last paragraph where it says:

25 "Sun puts Field of Use restrictions in the

1 Java SE TCK licenses which prohibit Java SE
2 implementations from running on anything but
3 a desktop or server. These restrictions
4 prevent Apache Harmony from independently
5 implementing Java SE."

6 Do you see that?

7 **A.** Yes.

8 **Q.** And do you recall Mr. Jacobs directed your attention to
9 that paragraph?

10 **A.** Yes.

11 **Q.** And what was the basis of that understanding that you were
12 sharing with Mr. Schmidt?

13 **A.** Let's see how far back I should go? So --

14 **Q.** Try and keep it short, but still understandable.

15 **A.** I will.

16 So the JCP is supposed to be and was -- was marketed
17 to be, I believe, Sun really intended for it to be an open
18 standards organization. That means that companies can come
19 together and work together and create these open standards that
20 anybody can implement and one company can't stop another
21 company from doing that.

22 And, in fact, the JSPA -- that's kind of the main
23 agreement that everybody signs when they participate in this
24 process -- explicitly prohibits them from doing so.

25 This went on -- this went great for quite awhile, and

1 there's Open Source projects that implement various
2 specifications, especially having the Java EE space --

3 **THE COURT:** Will you slow down some? I can tell the
4 court reporter's fingers are going way too fast. You need to
5 slow down some.

6 **THE WITNESS:** I certainly will.

7 **THE COURT:** You need to slow down.

8 **A.** This was going very well, especially, like, in the Java EE
9 space.

10 Up until the Apache Harmony project came along, which
11 tried to create a standard implementation of Java SE, that was
12 all going to go fine.

13 Actually, for the first couple years I think Sun just
14 didn't think that they could do it, so they didn't really --
15 nobody was really worried about it and they actually explicitly
16 said that it would be incredibly difficult to do this. But
17 then they did, in fact, do it or at least we think we did --
18 they did.

19 Harmony was able to create a complete implementation
20 of Java SE, is what I mean. And they went to submit it and
21 everything was going fine and, you know, they just assumed that
22 they would be able to get the same license as they got for the
23 Java EE specs, because this was -- there was already precedent
24 for this and there was already a process for this. But then at
25 the last minute Sun changed their mind and I -- the way they

1 have explained it to us in the JCP EC is that they were worried
2 about Harmony somehow creating competition with their -- the
3 licensing revenues they get for virtual machines running on
4 cell phones, just like -- and I'm talking about the little
5 feature phones. So kind of crazy they would be worried about
6 that, in my opinion.

7 But because of that, they used this loophole. As I
8 explained before, you have to -- they found and used this
9 loophole. As I explained before to, I guess, kind of get the
10 Java seal of approval for your -- for your implementation of a
11 spec is basically if I want to say that I'm a Java compatible
12 standard, I have to go through this process. I mean, it's like
13 a trademarked term --

14 **THE COURT:** This answer is way too long. This has
15 turned into a speech. You've got to give short answers.

16 And, counsel, please ask for a question that does not
17 call for a speech.

18 **MR. BABER:** Will do, your Honor.

19 **THE COURT:** Thank you.

20 **THE WITNESS:** Well --

21 **THE COURT:** No. Speeches are over. Next question.

22 **BY MR. BABER:**

23 **Q.** Mr. Lee, you just referenced in your answer what you
24 referred to as a "Java seal of approval." What did you mean by
25 that?

1 A. So, Java is a trademark. Like, I can't just make a new
2 technology and call it Java.

3 Q. Okay. And I want to go back in terms of Apache Harmony.
4 Who, if you know, was -- were the parties or entities who
5 formed Apache Harmony? Who caused Apache Harmony -- caused the
6 Apache Foundation, sorry, to come into existence in the first
7 place?

8 A. I -- I actually don't know who the founding members were.
9 I know that IBM was involved.

10 Q. Okay. Do you know whether the Apache Foundation was
11 itself a member of the Executive Committee of the JCP?

12 A. They were, yes.

13 Q. Do you know approximately when the Apache Foundation
14 joined the Executive Committee?

15 A. I think it was very early on. It was in the beginning.

16 Q. When you first became involved with the Executive
17 Committee, was the Apache Foundation already a member?

18 A. For a long time, yes.

19 Q. Is the Apache Foundation still a member?

20 A. No. They quit.

21 Q. Do you know when they quit?

22 A. It was about a year ago.

23 Q. Do you know why they quit?

24 A. Over this issue.

25 Q. Okay. Now, in the email to Mr. Schmidt, you are referring

1 to these restrictions that Sun was trying to impose.

2 Do you know, Mr. Lee, whether those restrictions were
3 ever, in fact, put in place with respect to Apache Harmony?

4 **A.** Yes. I was told so in the JCP EC meeting by both parties.

5 **Q.** Did Apache every sign the license agreement for the TCK?

6 **A.** No. They never adopted this license.

7 **Q.** And just so it's clear for the jury, Sun did -- did Sun
8 offer to the Apache Foundation a TCK license for Apache
9 Harmony?

10 **A.** Yes.

11 **Q.** And did or did not that license that Sun offered include
12 this Field of Use restriction?

13 **A.** They said it did, yes.

14 **Q.** Okay. And Apache Foundation did not accept that
15 restriction, is that right?

16 **A.** Correct.

17 **Q.** Mr. Lee, I would like to put back in front of you --

18 **MR. BABER:** May I approach, your Honor?

19 **THE COURT:** Yes.

20 **BY MR. BABER:**

21 **Q.** (Continuing) -- Exhibit 281, which was also shown to you
22 during your direct examination -- or your cross-examination, or
23 Mr. Jacobs' examination.

24 (Whereupon, document was tendered
25 to the witness.)

1 Q. Do you recall this email string?

2 (Document displayed)

3 A. Yes.

4 Q. Okay. And that includes an email from you at the top to
5 Hiroshi Lockheimer, is that right?

6 A. Yes.

7 Q. Who is Hiroshi Lockheimer?

8 A. He is the current director of the Android engineering
9 team.

10 Q. And it shows a copy to Dan Bornstein. Who was Dan
11 Bornstein in January of 2009?

12 A. He was the lead of the Dalvik team and my manager.

13 Q. Okay. And in the middle of that email chain, the message
14 in the middle shows "On Tuesday January 6, 2009 at 10:42
15 Hiroshi Lockheimer wrote."

16 Do you see the part I'm directing you to?

17 A. Yes, I do.

18 Q. Okay. And this is the part that Mr. Jacobs asked you
19 about. The second sentence says -- and he's referring to an
20 entity named Noser.

21 "Those guys (their management team) are super
22 shady."

23 Do you see that?

24 A. Yes.

25 Q. And then the next sentence goes to say:

1 "I know the engineers are great and I get
2 that sense, too, from my limited interactions
3 with them."

4 Who was Noser?

5 A. Noser is a company in Europe. I think they were based in
6 either Sweden or Germany. They had a history of working on
7 mobile VMs.

8 Q. And what's a VM?

9 A. It's a virtual machine.

10 Q. And a virtual machine is part of the software that allows
11 a language like Java to run on a computer, is that right?

12 A. Yes.

13 Q. Okay. And during your time as the core library lead for
14 Android, did you have interactions with software engineers from
15 Noser?

16 A. Yes.

17 Q. And what was Noser's role with respect to the Android core
18 libraries?

19 A. They played a supporting role and -- because it was a lot
20 of, I guess, rote work and they took on the less strategic
21 tasks.

22 Q. Okay. Now, in your dealings with Noser, did you have
23 occasion to work with the management team from Noser?

24 A. No, I did not.

25 Q. Do you have any idea what Mr. Lockheimer meant when he

1 said the management team from nose were, quote, super shady?

2 A. I know that later on they released their own virtual
3 machine. That's the only thing I know.

4 Q. But you did have interactions with the Noser engineers,
5 correct?

6 A. Yes.

7 Q. Did you find them to be professional?

8 A. I thought they were wonderful.

9 Q. Okay. Now, in the examination this morning, Mr. Jacobs
10 asked you about a couple different pieces of prose. English
11 language text that comes from the Android platform and from the
12 Java Platform, correct?

13 A. Yes.

14 Q. And first of all, what do you call -- what word do you use
15 to identify those parts of -- those paragraphs of text that you
16 were looking at?

17 A. Documentation, or in some cases specification.

18 Q. Do you use those two terms interchangeably "documentation"
19 and "specification"?

20 A. Yes. I think "documentation" is the more general term.
21 "Specification" refers to these specific rules that describe, I
22 guess, the meanings and how things behave. Those are just as
23 important as, I guess, the namings of things.

24 Documentation more generally means, include also
25 stuff like tutorials, and examples and that sort of thing.

1 Q. And what Mr. Jacobs showed you, the first one of them that
2 had a technical name about CipherInputStream, I believe you
3 told him that there were some words that are common.

4 And could you explain why you would expect some words
5 to be common when you're comparing two descriptions of the same
6 class or the same method in the Java APIs?

7 A. Right. Even outside of whether it's -- just in technology
8 there is very specific terms that mean very specific things.
9 You know, technology is a very specific thing and you have to
10 speak very specifically about it. So there is certainly very
11 common terms both in the wire technical community and in the
12 Java community.

13 For example, "cipher" is a very specific word. There
14 is not really another word that you can say besides cipher. It
15 means it's an algorithm that can encode and decode something.

16 Q. And when you are trying to write in English a description
17 of what a particular class or a method does, do you think you
18 could accurately describe, as a technical matter, what a
19 certain method does without using words that identify the input
20 to the method and what the method gives you back and what the
21 core function is?

22 MR. JACOBS: Your Honor, this is beyond leading at
23 this stage. And it's asking for an expert opinion by this
24 gentleman.

25 THE COURT: The latter objection is overruled, but

1 the leading objection is sustained.

2 **MR. BABER:** Yes, your Honor.

3 **BY MR. BABER:**

4 **Q.** What, if any, similarities would you expect to find in any
5 two descriptions of the same method or class within the Java
6 API libraries?

7 **A.** So these, especially the specification parts, are a
8 contract. These are specific rules. You know, I actually
9 wasn't even a big fan of including these. I would have
10 preferred that we just point people to Sun's site for this
11 specific documentation because you shouldn't really be
12 rewriting a contract. And in doing so they are going to be
13 substantially similar.

14 You're talking about a very specific thing and very
15 specific rules about technologies that have very specific
16 terminology, so there's only so many ways that you can phrase
17 things.

18 **Q.** Now, Mr. Lee, the Java Community Process that you
19 described earlier is the process through which people outside
20 of Sun, and now outside of Oracle, contribute things to the
21 Java Platform; is that correct?

22 **A.** Yes.

23 **Q.** Okay. Now, looking at any two descriptions of classes or
24 methods for the Java language APIs, can you tell just by
25 looking at the two of them whether one was copied from the

1 other?

2 **A.** No, I cannot.

3 **Q.** And why can't you tell that?

4 **A.** Because I have no idea where they came from. I don't know
5 who submitted that. They could have come from the same place.
6 One could have come from the other. You need, like, the commit
7 history and...

8 **Q.** What is the "commit history"?

9 **A.** So as we talked about, this is code. We use something
10 called a source control system that keeps track of changes to
11 that code and it enables you to go back to old code or just see
12 how things changed over time. So you would have a specific
13 history and see who submitted code at various times and,
14 hopefully, be able to tell where it came from.

15 As I said, in the Java Community Process there is
16 expert groups of people from all different companies that
17 create these specifications and so you have to, like, probably
18 look back at those mailing lists or something like that to see
19 who contributed that particular prose to find out who actually
20 owns it, I think.

21 **Q.** Okay. Now, it's been the case for many years that this
22 Java Community Process has been in place, correct?

23 **A.** I think since about 1999, yes.

24 **Q.** Okay. And when someone contributes something to the Java
25 Community Process and to the Java Platform, can that person who

1 wrote it in the first place also at the same time contribute it
2 to the public?

3 **A.** Absolutely.

4 **Q.** And make it free for everyone to use?

5 **A.** Yes.

6 **Q.** And can they also release --

7 **MR. JACOBS:** Your Honor, calls for a legal
8 conclusions and speculation that the witness has not
9 established that he knows these, the legal terms under which
10 specifications are contributed and owned.

11 **THE COURT:** That's true. He's not a lawyer.
12 Are you a lawyer?

13 **THE WITNESS:** No, but I was on the Executive
14 Committee --

15 **THE COURT:** Wait, wait.

16 You start saying things like "make it free for
17 everyone to use," that's a law conclusion. So the jury will
18 disregard that part of the witness's testimony.

19 Now, you may -- it would be okay for him to testify
20 if he has direct first-hand knowledge as to what the custom and
21 usage is. That would be okay, but he cannot testify about the
22 law.

23 So that objection is well taken. Sustained. Please
24 try it again.

25 **MR. BABER:** Thank you, your Honor.

1 BY MR. BABER:

2 Q. Are you familiar with the practices of some members of
3 the -- let's start with the Executive Committee of the JCP, as
4 to what their practice has been historically with respect to
5 how they have made available things that they are contributing
6 to Java?

7 A. I'm directly familiar, yes.

8 Q. And can you give us an example of what the practice of
9 some member of the Executive Committee has been with respect to
10 things that she or he have written and are at the same time
11 contributing to the Java Platform?

12 A. With the most -- my most recent specification that I
13 wrote, JSR 330, I released the same code under the Apache
14 license, which is very, very permissive Free license. Capital
15 "F" Free by that definition.

16 Q. Are there any other members of the Executive Committee
17 where you're familiar with his or her practices with respect to
18 what they have done in the past regarding code and new items
19 they have written for the Java Platform?

20 A. Particularly how they handled it or whether -- like how
21 they licensed it themselves?

22 Q. Just if you're familiar with any practice they had with
23 respect to contributions they were making to the Java platform.

24 A. I can think of several.

25 Q. Can you -- do you know of any examples of someone who, at

1 the same time they donated something to the Java platform, also
2 made it available under either a license or some mechanisms --
3 let me back up and lay a foundation.

4 As a software engineer and familiar with the Java
5 platform in particular, Mr. Lee, are you familiar with some of
6 the public licenses that are available in the software -- in
7 the software world?

8 **A.** Yes.

9 **Q.** What are some of the public licenses you're familiar with?

10 **A.** The Apache license. The variation of the VSE license.
11 The GPL. The MIT license. The X11 license. The Mozilla
12 license. There's a lot of them.

13 **Q.** Just as a general matter -- I'm not asking for any legal
14 construction -- but as a software engineer and someone who
15 served on the JCP, what is your general understanding of a,
16 quote, public license to software?

17 **A.** Uhm, so, for open source licenses, that basically means
18 that you could take the code and reuse it in various ways that
19 are specified by the license.

20 Some require you to give attribution. Some say
21 that -- for example, the GPL says that if you open source -- or
22 if you modify their code, then you also have to open source
23 your changes if you release that code.

24 **Q.** And these licenses -- again, your understanding as a
25 software engineer and someone who has been involved in the Java

1 Community Process -- can those be true, can they be applicable
2 with respect to API packages, for example?

3 **A.** With respect to --

4 **MR. JACOBS:** Your Honor, relevance.

5 **THE COURT:** Yes. Well, was this witness designated
6 as an expert?

7 **MR. BABER:** No, Your Honor. He's a fact witness who
8 served on the JCP, executive committee.

9 **THE COURT:** You know, the other day you raised an
10 objection that the other side had not given the Rule 26
11 disclosure about somebody was going to give expert testimony or
12 opinion testimony. Didn't I hear that a few days ago?

13 **MR. BABER:** I'm trying to remember who that witness
14 was, Your Honor.

15 **THE COURT:** I think you've gotten into the point
16 where this witness is now giving specialized testimony that
17 should have been noticed under the rules. If it wasn't
18 noticed, I'm going to sustain that objection.

19 **MR. BABER:** Your Honor, may I ask him, as matter of
20 fact, certain things?

21 **THE COURT:** No. These are not facts. These are
22 specialized experience. That's beyond the can of the ordinary
23 juror. This is what we normally cover as expert testimony.

24 **BY MR. BABER:**

25 **Q.** Mr. Lee, back to the issue of the documentation that we

1 were speaking about earlier, you indicated that in order to try
2 to form an assessment as to whether one paragraph of English
3 text had been copied from another paragraph of English text,
4 you would need more information.

5 **A.** Yes.

6 **Q.** Okay. And you mentioned earlier at least one source you
7 would go to, which was --

8 **A.** Source Control and then Mailing Lists.

9 **Q.** Would there be any sources outside of, let's say, Google
10 that you would want to check to look into the history of the --
11 that paragraph and where it had come from?

12 **A.** It would depend where that code came from. So, yeah,
13 certainly, IBM, Intel. I don't know who contributed ...
14 there's another open source project called Bouncy Castle, that
15 Android uses and is popular in the Java community.

16 **Q.** What is Bouncy Castle?

17 **A.** It's an implementation of those crypto libraries we saw.

18 So the cypher is just a very abstract type in the
19 Java spec itself. It doesn't do anything. It just says
20 there's such a thing exists as an algorithm that can code and
21 decode things. If you want to actually encode and decode
22 things, you need an implementation.

23 And Bouncy Castle is a common open source one and an
24 implementation that Android uses.

25 **Q.** And in connection with the questioning that Mr. Jacobs did

1 this morning, where he showed you those couple of paragraphs,
2 were you provided with any information about where those
3 paragraphs had originally come from and who had written them?

4 **A.** I was not.

5 **Q.** Okay. And without that information, is it possible to
6 make a determination as to whether one of them was copied, the
7 other one was copied, or perhaps they were both copied from
8 somewhere else?

9 **A.** No.

10 **Q.** Now, Mr. Lee, during your time -- Mr. Jacobs asked you,
11 during his examination last week, whether when you were
12 reviewing the Javadocs specifications for the API packages,
13 whether or not you saw a copyright notice on those materials.

14 Do you recall that?

15 **A.** Yes.

16 **Q.** Okay. Now, during the time you were involved in the JCP
17 in connection with the issues relating to Apache Harmony, did
18 you ever hear anyone from Sun or Oracle ever say that Apache
19 Harmony had ever infringed any copyrights of Sun or Oracle?

20 **A.** No.

21 **Q.** And when you were doing your work at Google on the Java
22 core libraries for Android and you were involved with the JCP,
23 did you obtain any information through the JCP process
24 regarding what Sun's view was of whether others were free to
25 use the API specifications?

1 **MR. JACOBS:** Your Honor, calls for hearsay.

2 **MR. BABER:** Your Honor, would be an admission.

3 **MR. JACOBS:** Double hearsay.

4 **THE COURT:** That's true. That's true. This question
5 calls for some comment that somebody else attributed to Sun.
6 And that would be hearsay. Sustained.

7 **MR. BABER:** Your Honor, can we limit it just to
8 things that people from Sun themselves said, rather than what
9 other people might have said about Sun comments? Anything Sun
10 said would be an admission, Your Honor.

11 **THE COURT:** Anything that someone authorized to speak
12 on behalf of Sun. So you would have to establish that the
13 person -- for example, if the representative from Sun on this
14 particular committee made a comment, okay, that would be
15 allowed.

16 **MR. BABER:** Thank you, Your Honor.

17 **BY MR. BABER:**

18 **Q.** Mr. Lee, in the course of your involvement with the JCP,
19 did you ever hear any statements by the representatives of Sun
20 to the JCP about whether other parties could do implementations
21 of the API specifications?

22 **A.** Uhm, I mean, certainly, that's the whole point of the JCP.

23 **THE COURT:** Ah, ah, ah, ah, ah. Come on.

24 **THE WITNESS:** I guess --

25 **THE COURT:** You're just making an argument. You're

1 giving a speech.

2 **THE WITNESS:** I just don't understand.

3 **THE COURT:** The question is, did you hear a specific
4 statement by that person? Then you can tell us what that
5 statement was.

6 But you can't say, well, I was sitting there, and I
7 drew an inference, and I was looking out the window, and I
8 think this is what he was talking about, but I don't really
9 know. That's not the whole point of it. That's not the
10 question.

11 **THE WITNESS:** Got it.

12 **THE COURT:** If there was a specific statement, that's
13 what we want to hear.

14 **THE WITNESS:** About -- what was the statement in
15 reference to, again?

16 **BY MR. BABER:**

17 **Q.** About parties doing their own implementations of the Java
18 APIs.

19 **A.** There's --

20 **Q.** Yes --

21 **A.** There was a blog post from Jonathan Schwartz, for example,
22 that congratulated Android.

23 **THE COURT:** Was he on this committee?

24 **THE WITNESS:** No.

25 **THE COURT:** All right. The jury is going to

1 disregard that speech.

2 This witness is making a lot of speeches for some
3 reason, Mr. Baber. We do not want speeches. The question
4 called for what? The guy on the committee.

5 Did the guy on the committee or the woman on the
6 committee make that statement?

7 **THE WITNESS:** I don't recall specific things Sun
8 representatives said on the committee.

9 **THE COURT:** All right. He doesn't recall. Thank
10 you. Next question.

11 **BY MR. BABER:**

12 **Q.** During the time you were at Google and working on the core
13 libraries and looking at the Javadocs, did you have any
14 conversations with anyone else at Google about making use of
15 the Javadocs in connection with the creation of the core
16 libraries?

17 **A.** Questions about like making use of the Java -- oh, of the
18 original Sun Javadocs?

19 **Q.** Yes.

20 **A.** Yes, we definitely referred back to them so that we could
21 make sure that we were maintaining -- not -- or maintaining
22 interoperability with their implementations.

23 **Q.** Okay. Did you have any conversations with anyone at
24 Google about whether it was appropriate for you to refer to the
25 Javadocs?

1 **MR. JACOBS:** Your Honor, just to caution, he is
2 opening the door.

3 **BY MR. BABER:**

4 **Q.** Okay. Let me make clear, I don't want any testimony about
5 any discussion you may have had with any attorneys. I'm only
6 talking about engineers or business people at Google.

7 **MR. JACOBS:** Same caution, Your Honor.

8 **THE COURT:** Maybe a door is being opened. I don't
9 know. I'm not even sure I know what you're talking about, but
10 counsel apparently does.

11 So there's no objection, so you ask whatever question
12 you want. If some door is open, well, then, we all walk
13 through it.

14 **MR. BABER:** Okay.

15 **THE COURT:** Go ahead.

16 **THE WITNESS:** I didn't know if this sort of thing was
17 allowed, so I asked my director of engineering, Steve Horowitz,
18 whether if you were allowed to reimplement other APIs like
19 this.

20 **BY MR. BABER:**

21 **Q.** What was your conversation with Mr. Horowitz?

22 **A.** He said that, yes, there's lots of precedent for this;
23 that you -- that's what we're talking about when you create a
24 clean-room implementation. And he gave me some examples.

25 **Q.** And, again, Mr. Horowitz is an engineer?

1 A. He's a former engineer and now manager.

2 Q. Okay. And how did he relate to you in sort of the pecking
3 order at Google? Was he --

4 A. He was my manager.

5 Q. He was your manager?

6 A. At the time, yes.

7 MR. BABER: Thank you, Your Honor. No more
8 questions.

9 THE COURT: Redirect examination.

10 REDIRECT EXAMINATION

11 BY MR. JACOBS:

12 Q. Mr. Lee, you feel very strongly about the Apache Harmony
13 dispute; don't you?

14 A. Uhm, yes. I'm very passionate about it.

15 Q. And if you were speaking at length about it in this
16 courtroom it's because you feel so strongly about it; isn't it,
17 sir?

18 A. Yes, sir.

19 Q. And it was a big fracas in the industry; wasn't it?

20 A. Yes, sir.

21 Q. Companies lining up on one side, trying to persuade Sun to
22 allow its property to be used in a way that Sun didn't want it
23 to be used; correct, sir?

24 A. To persuade Sun to adhere to the agreement we had all
25 signed, yes.

1 Q. Well, that's interesting, sir; isn't it? No one had ever
2 filed a lawsuit saying Sun was violating the agreement; did
3 they?

4 A. No.

5 Q. Apache never initiated legal action saying, Sun, you
6 violated the JSPA; did it?

7 A. They didn't have the resources, no.

8 Q. And, in fact -- well, IBM was involved in Apache.

9 A. I'm sorry?

10 Q. Wasn't IBM involved in Apache Harmony?

11 A. Yes, they were.

12 Q. Now --

13 A. But IBM is also a partner with Sun.

14 This issue was certainly discussed quite a bit,
15 whether that was a viable --

16 Q. I would like you to take a look at Exhibit 1047.

17 MR. JACOBS: May I, Your Honor?

18 THE COURT: Yes.

19 THE WITNESS: Thank you.

20 BY MR. JACOBS:

21 Q. 1047 is an open letter to Sun Microsystems from the Apache
22 Software Foundation, and a FAQ open letter to Sun Microsystems.
23 Do you see that?

24 A. Yes.

25 Q. Actually, to be more precise, it is the FAQ portion of

1 this document; isn't it, sir?

2 **A.** Yes, sir.

3 **Q.** And you recognize it?

4 **A.** Yes.

5 **MR. JACOBS:** Your Honor, 1047 moved into evidence,
6 please.

7 **MR. BABER:** No objection.

8 **THE COURT:** 1047 received.

9 (Trial Exhibit 1047 received in evidence.)

10 **BY MR. JACOBS:**

11 **Q.** And I would like you to turn to the page marked 6 of 10.

12 **A.** Do you mind if I just read on the screen?

13 **Q.** That will be fine.

14 **A.** Okay.

15 **THE COURT:** Does the jury have that in the jury box?

16 **MR. JACOBS:** It's being posted, yes, Your Honor.

17 **BY MR. JACOBS:**

18 **Q.** You see there's a question in the middle there, says, "Why
19 doesn't Apache simply ignore this" -- referring earlier to all
20 the disagreements -- "and ship Harmony without passing the
21 JCK?"

22 Do you see that?

23 **A.** Yes.

24 **Q.** And the answer on this FAQ from Apache Harmony is as
25 follows:

1 "We can ship Harmony without passing the JCK.

2 It's our source code to do with what we wish,
3 and we will with milestone releases as we
4 progress towards completion.

5 "However, we could never claim to be Java
6 compatible, which is something very important
7 to Java users, and is the stated goal of the
8 project.

9 "Also, users wouldn't be assured that they've
10 had all necessary IP rights from the spec's
11 contributors. Compatibility is important to
12 us, as is not putting users in IP jeopardy as
13 it has been for every JSR the ASF has ever
14 implemented."

15 Then it concludes:

16 "We have no interest in forking the
17 technology."

18 Do you see that, sir?

19 **A.** Yes.

20 **Q.** And this is a document that was created in the -- during
21 this big fracas between Apache and its members and Sun;
22 correct, sir?

23 **A.** Yes.

24 **Q.** Now, I'd like you to take a look at another document.

25 This will be 1045.

1 **MR. JACOBS:** May I, Your Honor?

2 **THE WITNESS:** Thank you.

3 **BY MR. JACOBS:**

4 **Q.** You referred in your testimony, sir, to the fact that
5 Apache software foundation resigned from the JCP executive
6 committee.

7 Do you recall that testimony?

8 **A.** Yes, sir.

9 **Q.** And this is a posting on the Apache Software Foundation
10 blog about that resignation on December 9, 2010. Do you see
11 that?

12 **A.** Yes.

13 **Q.** And you recognize this posing; don't you, sir?

14 **A.** I probably read it at the time.

15 **MR. JACOBS:** Your Honor, 1045 in evidence.

16 **MR. BABER:** No objection.

17 **THE COURT:** Received.

18 (Trial Exhibit 1045 received in evidence.)

19 (Document displayed.)

20 **BY MR. JACOBS:**

21 **Q.** I would like to direct your attention, Mr. Lee, to the
22 third to last paragraph of the document.

23 **A.** Okay.

24 **Q.** By the way, Apache was furious over this, right?

25 **A.** I don't know that they really had -- I wouldn't really

1 attribute emotion to it like that, but it was certainly
2 disappointing. I would consider them very objective through
3 the whole matter, actually.

4 Q. A strong business disagreement over Sun's policy, and
5 later Oracle's policy; correct, sir?

6 A. "Strong disagreement" is a good way to put it.

7 Q. So there's a lot of material in this blog posting that
8 sets forth Apache's side of this disagreement about Sun's
9 intellectual property; correct, sir?

10 A. I guess so, yes.

11 Q. And in this third-to-last paragraph of Exhibit 1045,
12 Apache writes:

13 "The Apache Software Foundation concludes
14 that" -- there's a typo -- "that the JCP is
15 not an open specification process; that Java
16 specifications are proprietary technology
17 that must be licensed directly from the spec
18 lead under whatever terms the spec lead
19 chooses."

20 Do you see that?

21 A. Yes.

22 Q. And so Apache concluded that these specifications that
23 we're talking about for the APIs were proprietary technology;
24 didn't they, sir?

25 MR. BABER: Objection, Your Honor. Foundation for

1 this witness.

2 **THE WITNESS:** That they --

3 **THE COURT:** Wait a minute. All right. It's true --

4 this is within the same latitude that, Mr. Baber, you used.

5 So, yes, your objection is well-taken. But you asked questions

6 of a similar character, and I'm going to allow responsive

7 questions up to the point that I said no more expert testimony.

8 So I will allow this question. Overruled. Go ahead

9 and answer the question.

10 **THE WITNESS:** Will you please repeat the question.

11 **MR. JACOBS:** Could you read it back, please.

12 (The reporter read the pending question.)

13 **BY MR. JACOBS:**

14 **Q.** That calls for a "yes" or "no," Mr. Lee.

15 **A.** Java specifications are proprietary. Yes, that's what

16 this sentence says.

17 **Q.** And Apache concluded that a license was required from the

18 spec lead under whatever terms the spec lead chooses; did they

19 not, sir?

20 **A.** That's the unfortunate reality, yes.

21 **Q.** Now, you talked about your conversation with Mr. Horowitz,

22 about the permissibility of what you were doing in

23 reimplementing the Java core library APIs. Do you recall that

24 testimony?

25 **A.** Yes.

1 Q. Did Mr. Horowitz tell you that he had consulted any
2 counsel at Google, in giving you this kind of advice about the
3 legal permissibility of reimplementing these APIs?

4 MR. BABER: Objection, Your Honor. Privilege.
5 Should be a yes or no answer.

6 MR. JACOBS: Opened the door, Your Honor.

7 THE COURT: Opened the door, certainly, to that
8 question. So, overruled. Go ahead answer the question.

9 THE WITNESS: Not that I recall.

10 BY MR. JACOBS:

11 Q. So Mr. Horowitz was giving you un- -- advice that, as far
12 as you knew, was unguided by any legal counsel; is that
13 correct, sir?

14 A. Correct.

15 MR. JACOBS: No further questions.

16 THE COURT: All right. Anything more?

17 MR. BABER: Very briefly, Your Honor.

18 I would like to have the same Exhibit back up in
19 front of Mr. Lee, number 1045, please. And back to that same
20 paragraph towards the bottom, that Mr. Jacobs asked about.

21 (Document displayed.)

22 RECROSS EXAMINATION

23 BY MR. BABER:

24 Q. Do you see in that -- right after the part that Mr. Jacobs
25 asked you about, Mr. Lee, the next phrase says, and this is

1 from the intro:

2 "The Apache Software Foundation concludes" --
3 and I'm reading -- "that the commercial
4 concerns of a single entity, Oracle, will
5 continue to seriously interfere with and bias
6 the transparent governance of the ecosystem."

7 Do you see that?

8 **A.** Yes.

9 **Q.** And what ecosystem is that referring to?

10 **A.** This is the Java Community Process.

11 **Q.** And if you go back to the top of the exhibit, very first
12 page, the date of this blog -- if I'm reading correctly -- is
13 December 9, 2010; is that right?

14 **A.** Yes.

15 **Q.** And that was after Oracle filed this lawsuit; isn't it,
16 Mr. Lee?

17 **A.** Yes.

18 **MR. BABER:** No further questions.

19 **MR. JACOBS:** Nothing further, Your Honor.

20 **THE COURT:** May Mr. Lee be excused and discharged,
21 not subject to recall?

22 **MR. JACOBS:** Yes, Your Honor.

23 **MR. BABER:** We may bring him back in our case.

24 **THE COURT:** All right. Well, at least you're
25 discharged from the subpoena. So, thank you, sir. You're free

1 to go. Have a good day.

2 **THE WITNESS:** Thanks, Your Honor.

3 Do I leave these papers here?

4 **THE COURT:** Yes. Take care.

5 **THE WITNESS:** Thanks.

6 **THE COURT:** I think we should start the next witness.

7 It's early for the mid-morning break, unless my jury needs a
8 break. Do any of you need a break? We'll go another 15 to 20
9 minutes. Next witness, please.

10 **MR. JACOBS:** Your Honor, Oracle calls Professor John
11 Mitchell.

12 **THE COURT:** All right. Will he please come forward.

13 **MR. JACOBS:** Your Honor, we've conferred, and Google
14 has no objection to Professor Mitchell having his opening
15 expert report in front of him on the stand, for reference.

16 **THE COURT:** All right. Great.

17 While we're waiting -- where is our expert? Okay.
18 He's being found. While he's coming forward, let me give you a
19 heads up over there.

20 You're going to hear a lot about the expert report.
21 And I've learned the hard way that juries think the expert
22 report will be in evidence in the jury room. No. It's a
23 hearsay document. And it's just a requirement, an important
24 requirement for the experts before they testify here. But they
25 typically do not come into evidence.

1 So if there's something that you hear that -- or some
2 diagram or something that you feel is important, make a note of
3 it rather than rely upon the erroneous assumption that you
4 don't need to because it will be in the jury room later on.

5 All right. That's just a heads up. It's always up
6 to you to decide how many notes to take.

7 Are you Professor Mitchell?

8 **THE WITNESS:** Yes, I am.

9 **THE COURT:** Welcome. Please raise your right hand.

10 **JOHN MITCHELL,**

11 called as a witness for the Plaintiff herein, having been first
12 duly sworn, was examined and testified as follows:

13 **THE WITNESS:** I do.

14 **THE CLERK:** Okay. Thank you.

15 **THE COURT:** Thank you. Welcome, again.

16 You've got to be about this close to the microphone.
17 And it will move all around, including the base.

18 **THE WITNESS:** Right.

19 **THE COURT:** Lean forward and then figure it out. Why
20 don't you say your name.

21 **THE WITNESS:** John Mitchell.

22 **THE COURT:** Not quite loud enough.

23 **THE WITNESS:** Not loud enough.

24 **THE COURT:** That was loud enough.

25 **THE WITNESS:** All right. John Mitchell.

1 **THE COURT:** Can you all hear?

2 **THE WITNESS:** Will that work?

3 **THE COURT:** All right. If any of you don't hear,
4 raise your hand at some point.

5 Thank you.

6 Please go ahead, counsel.

7 **DIRECT EXAMINATION**

8 **BY MR. JACOBS:**

9 **Q.** Professor Mitchell, could you tell the jury where you work
10 and what your position is.

11 **A.** I work at Stanford University. I teach computer science
12 in the Computer Science Department there.

13 My official title is the Mary and Gordon Crary Family
14 Professor in the School of Engineering.

15 **Q.** Can you describe your educational background that led you
16 to end up at Stanford.

17 **A.** Sure. I was an undergraduate at the University of
18 Wisconsin in Madison, Wisconsin. I then transferred to
19 Stanford, where I graduated with a degree in math.

20 After working for a couple of years, I then went to
21 graduate school in Boston, at MIT. I got a master's and Ph.D.
22 in computer science from MIT.

23 **Q.** Then what did you do before actually taking a teaching
24 position at Stanford?

25 **A.** I worked at the research lab called Bell Laboratories of

1 AT&T, for about four years. And then I was offered a job in
2 beautiful California, and I've been here since.

3 **MR. JACOBS:** Your Honor, may I approach?

4 **THE WITNESS:** Sure.

5 **MR. JACOBS:** I have handed the witness Exhibit 687.

6 **BY MR. JACOBS:**

7 **Q.** Can you tell us what 687 is.

8 **A.** This is what's called a curriculum vitae. It's a summary
9 of my professional career.

10 **MR. JACOBS:** Offer 687 into evidence.

11 **MR. VAN NEST:** No objection, Your Honor.

12 **THE COURT:** 687 received.

13 (Trial Exhibit 687 received in evidence.)

14 **BY MR. JACOBS:**

15 **Q.** Tell us a little bit about the courses you teach at
16 Stanford.

17 **A.** Every fall I teach a course in programming languages.
18 I've done that for about 20 years. I also teach courses on
19 computer security in the winter and spring. And sometimes in
20 the winter also an advanced course on programming languages.

21 **Q.** Do any of your classes cover the Java programming
22 environment?

23 **A.** Yeah. My regular fall course has a section of Java. I
24 wrote a textbook for that course, and the textbook has a
25 chapter on Java.

1 Q. Have you done research related to Java?

2 A. Yes. A number of different kinds of research. One line
3 of research involved exploring type systems, ways of checking
4 properties of programming languages.

5 I wrote a series of papers with a former student,
6 looking at principles and mechanisms that were then
7 incorporated into the Java programming languages.

8 Q. So can you just explain that last bit a little more.
9 Something was incorporated into the --

10 A. One of the main revisions or extensions to Java involves
11 something called Generics. And I think we'll see that in some
12 of the sample code, if we look at that.

13 Q. And the -- your curriculum vitae, does it have a list of
14 your publications?

15 A. Yes, it does.

16 Q. Did you have any prior relationship with Oracle before you
17 were engaged to help on this litigation?

18 A. No, I did not.

19 MR. JACOBS: Your Honor, I would offer Professor
20 Mitchell as a qualified expert in the area of computer science
21 and programming languages.

22 MR. VAN NEST: No objection, Your Honor.

23 THE COURT: All right. So long as the witness stays
24 within his field of expertise, there will be no issue. So,
25 please, go right ahead.

1 **BY MR. JACOBS:**

2 **Q.** Now, Professor Mitchell, I want to walk through some
3 information to help the jury understand the APIs that are at
4 issue in this lawsuit.

5 In order to do that, I would like to rely on some
6 things that have been shown to the jury and described already
7 in court. So up here on the top was a -- kind of a map of an
8 application programming interface and a package that implements
9 that interface.

10 Do you recall learning about this, this poster?

11 **A.** Yes, I do.

12 **Q.** And this is --

13 **THE COURT:** May I suggest that you pull it over.

14 Mr. Van Nest, do you mind if we pull it closer to the
15 jury so the jury can -- it's a little far for the jury.

16 **MR. VAN NEST:** Of course not, Your Honor.

17 **THE COURT:** So -- there we go. That better? Can you
18 all see it at that distance? Excellent. All right. Go ahead.

19 **BY MR. JACOBS:**

20 **Q.** That's Trial Exhibit 3452. And then the jury also heard a
21 description of the Java Class Libraries poster, which is Trial
22 Exhibit 1028. And you're familiar with both of these, correct?

23 **A.** I'm familiar with the poster.

24 **Q.** So now let's go to some slides that you prepared.

25 So this is the -- this is TX 1028, the Java poster.

1 Review with us what this poster is showing.

2 **A.** This is a great poster. This shows a number of packages
3 from the Java Class Library. For each package there's a really
4 nice, graphical notation showing the relationships between
5 elements in that package; and, also, there's a color-coding
6 system used to show connections and relationships between
7 separate packages.

8 **Q.** What's the basic concept in the Java class libraries
9 illustrated on the poster?

10 **A.** The most basic concept is a class. And a class is used in
11 object-oriented programming to create objects. Each object
12 created by a class has the methods. Those are operations on
13 objects of that class. And fields, each object has fields
14 which are, again, associated with objects of the class.

15 **Q.** Can classes be related to each other?

16 **A.** Yes, they can. The most fundamental relationship between
17 classes is the subclass relationship. When one class has all
18 of the methods and fields of another, it can be declared to be
19 a subclass of the first class.

20 **Q.** So let's look at a portion of this poster. We're looking
21 at the lower right-hand corner, down here, of this exhibit.
22 Can you tell us what this is showing?

23 **A.** This is the graphical notation. To me, it kind of reminds
24 me of musical notation because there are lines and dots in it.

25 On the left side there are classes. And here there's

1 a vertical line underneath a class with horizontal solid lines
2 connecting to subclasses of that class. So that illustrates
3 the class/subclass relationship that's fundamental to
4 object-oriented programming.

5 Another relationship illustrated here is a
6 relationship between classes and interfaces. An interface
7 lists a set of methods. And a class is related to that
8 interface if the class provides all the methods and other
9 elements listed in that interface.

10 Here, this word "interface" is used in a particular
11 way specific to the Java language. It's different from the
12 more generic use of the word "interface" in the term API or
13 application program interface.

14 **Q.** How are the classes and interfaces grouped?

15 **A.** Uhm, the classes are hierarchical, and grouped under each
16 other. The interfaces can be hierarchical. Classes,
17 interfaces, and other elements can be organized into packages.
18 And packages, again, can be hierarchical. A package can have a
19 number of subpackages that are related to it in some way.

20 Another thing that's shown on this legend for the
21 poster is the way a class might -- in one package, might be a
22 subclass of a class that comes from another package. So that's
23 another kind of relationship between packages and classes.

24 Here, this illustration shows a class with a blue dot
25 with a star in it. That's to indicate that that class in the

1 package that's being shown comes from another package.

2 Since packages are hierarchical, the package that one
3 class comes from might, in fact, be a subpackage of another
4 package. And the numbering system is used in a kind of
5 succinct way to illustrate that.

6 **Q.** Is there a rule of thumb about how many classes are
7 typically contained in one of the API packages illustrated on
8 this poster?

9 **A.** There's no absolute limit. A package can have a small
10 number of classes or a large number of classes. For the
11 packages that are at issue in this case, there's an average of
12 maybe 10 or 12 classes per package. But, it varies.

13 **Q.** Let's look at an example of one of the APIs that is on
14 the -- in the packages, the 37 packages that are at issue in
15 the lawsuit.

16 We're showing you java.net on slide 3 of your
17 demonstratives, Dr. Mitchell. Can you tell us what this --
18 zooming in on java.net shows us?

19 **A.** This is an example of one of the packages shown on the
20 poster. There's a list of classes in the left column. And the
21 indentation and lines in the left shows the subclass
22 relationship.

23 Sometimes that class -- subclass relationship is two
24 or three classes deep. One example, URLs are something maybe
25 people are familiar with. Uniform Resource Locator. This is

1 an address, like Amazon.com or Facebook.com, that you type into
2 a web browser.

3 Underneath URL class there's another class, called
4 HttpURLConnection. This is used for networking, to connect to a
5 URL. And there are a couple of different subclasses under
6 that.

7 The interfaces in the right column are also
8 hierarchical. In principle, to make things a little bit
9 simpler to fit on the poster, they are just listed in a
10 vertical line, vertical column. But they could be indented to
11 show a subinterface relationship, as well.

12 To give one example, the URL class supports the
13 Serializable interface. So there's a dotted line from "URL"
14 going over to a vertical dotted line, and that connects up to
15 the Serializable interface.

16 Another thing I might point out is that some of the
17 classes defined in java.net are subclasses of classes that are
18 defined in another packages.

19 Towards the bottom there's a class called
20 URLClassLoader. That's a subclass of something called
21 SecureClassLoader that comes from another package. And then
22 SecureClassLoader is, in turn, a subclass of ClassLoader, which
23 comes from third package.

24 So there's many kinds of relationships between
25 classes, interfaces, and packages. Not only are they

1 hierarchical, but there are other forms of connections, such as
2 a class and a subclass of another class from another package.

3 ClassLoader class, for example, has this little blue
4 icon there. I think that refers to java.util. So if you
5 wanted to figure out where the ClassLoader class is, you would
6 look in that other part of the chart or poster and see how that
7 sits in the hierarchy there.

8 **Q.** Are methods shown on this description of java.net?

9 **A.** No. In order to make everything fit on the poster, the
10 methods are left out.

11 **THE COURT:** Where would they go if they were shown?

12 **THE WITNESS:** I think the logical place to show them
13 would be listed under each class, because a class is
14 characterized by the set of methods and fields associated by
15 that class.

16 **THE COURT:** Take where it says "Authenticator," the
17 very first one.

18 **THE WITNESS:** So, Authenticator may have some
19 methods, and you could list them under that. I think we have
20 another illustration later, that uses that idea.

21 **THE COURT:** All right. Thank you.

22 Is there a way to below that thing -- the one you
23 have -- no, no, the -- is the one you just showed us on the
24 screen a blowup of one of the --

25 **THE WITNESS:** Yeah, that's a piece of the poster,

1 Your Honor.

2 **THE COURT:** I didn't understand that. Thank you.

3 **BY MR. JACOBS:**

4 **Q.** So now we're zooming in on something called java.io.

5 Is that another package that is among the 37 packages
6 in dispute, Dr. Mitchell?

7 **A.** Yes, it is. The little magnifying glass on the left is
8 meant to indicate that this java.io illustration is a
9 magnification of the java.io portion of this larger poster.

10 **Q.** And then we've added in another package, "java.util." Do
11 you see that on the screen?

12 **A.** Yes, I do.

13 **Q.** Is java.util another package that's one of the 37 in
14 dispute?

15 **A.** Yes, it is.

16 **Q.** And then, finally, java.nio -- and the jury has heard a
17 lot about .nio and how it got developed -- is that another one
18 of the packages in dispute?

19 **A.** Yes, it is.

20 Most of the packages in dispute are shown on this
21 poster. Some are not. The poster also includes packages that
22 are not in dispute.

23 **THE COURT:** What would that be? How many are -- how
24 many are on this document?

25 **THE WITNESS:** I don't remember the exact number. I

1 think they are on the order of 10 to 15 that are not of the 37.
2 There's a few that are not here.

3 **THE COURT:** All right.

4 **THE WITNESS:** And some the other direction.

5 **THE COURT:** So like java.io, that's an API? Is that
6 an API?

7 **THE WITNESS:** Technically, it's a package. That's a
8 grouping of classes and interfaces. The API consists of
9 information about packages, classes, and interfaces, together
10 with a description of what these things mean.

11 So this is a -- an outline or an overview of a
12 portion of the API.

13 **THE COURT:** Okay. Thanks.

14 **BY MR. JACOBS:**

15 **Q.** So, now, we're going to zoom in a little closer on
16 java.nio. And what are we showing at this portion of the
17 demonstrative, Dr. Mitchell?

18 **A.** So here's an expansion of the poster, following up on the
19 idea of listing the methods, as well.

20 Here's a list of the methods for the ByteBuffer
21 class. And in each line showing a method there's the method
22 name and the type of parameter or parameters that the method
23 takes, and then the return type of the method.

24 So method is an operation that does something. In
25 order to perform the operation, a programmer might need to

1 supply additional data. And then when the operation is done
2 return results, some value might come back, some data might
3 come back from the method.

4 So the one that's in a little red box, there is a
5 getMethod. A byte buffer is like a list of bytes in order.
6 And if you wanted to get the third byte out of a ByteBuffer,
7 you would call the get method with an argument 3. And then
8 that method would return the third byte as a result of
9 performing that operation.

10 **Q.** Is there the concept of inheritance illustrated on this
11 slide, Dr. Mitchell?

12 **A.** The subclassing relationship is associated with what's
13 called inheritance. If in the programming language you define
14 one class to be a subclass of another, then the subclass
15 inherits all the methods of the superclass, the opposite of
16 subclassing.

17 I think there's another list here that shows the
18 methods that ByteBuffer inherits from Buffer.

19 **Q.** Can you just explain what we're now showing on the screen
20 in a little more detail?

21 **A.** This list is just the list of method names for Buffer.
22 Because ByteBuffer is declared to be a subclass of Buffer,
23 every operation or method from the Buffer class is
24 automatically inherited to ByteBuffer, and can be used as an
25 operation on any byte buffer.

1 Q. Now, let's look at the relationship between application
2 programming interfaces and class libraries. So what are we
3 illustrating on the screen at slide 5?

4 A. On the left-hand side of this screen there's a class
5 "Channels" and a list of some methods associated with Channels.

6 So the API will contain that class, descriptions of
7 these methods. And then the class library itself has the
8 method names and their types associated with them, together
9 with executable code that performs the operations associated
10 with these methods.

11 So each of the red boxes there, I think, represents
12 some amount of executable byte code. The amount depends on the
13 complexity of the method. That could be long or short
14 depending on how much work is involved in implementing or
15 running that operation on the computer when a program calls it.

16 Q. Sorry. If we go back to this poster, 3452, and we look at
17 the material that's above the black line on the poster, that's
18 identified as "name," so "java.lang.map public static int max
19 (int arg 1, int arg 2)," do you see that?

20 A. Yes.

21 Q. What is this in the application programming interface
22 (indicating)?

23 A. There's a part in the box that's like the declaration for
24 max. And above that is descriptive information saying how that
25 fits in the package.

1 Q. So the declaration is: "Public static int max (int arg 1,
2 int arg 2)." Correct?

3 A. That's correct. That says there's a method called max.
4 For some reason, the method name is in the middle, lost in a
5 lot of other things.

6 The modifiers or terms/words to the left say how that
7 sits in the class. And after the parentheses, there's the
8 parameters, the things that a programmer -- kinds of things
9 that a programmer has to supply to that operation in order for
10 the code to execute and complete the operation named --

11 Q. Now --

12 A. -- on the method.

13 Q. Now, on this slide, what you said was illustrated was the
14 class library in its binary form; is that right, Dr. Mitchell?

15 A. That's correct.

16 Q. So how, in binary form, in the actual class library
17 itself, would this declaration be represented?

18 A. That's -- there's a class file format. And information
19 about that declaration is represented in text form in the class
20 file, in the constant pool portion of that.

21 (Reporter interrupts.)

22 **THE WITNESS:** Constant pool. Like swimming pool.

23 **BY MR. JACOBS:**

24 Q. Below the declaration line on 3452, there is additional
25 source code. Do you see that?

1 A. Yes, I do.

2 Q. And how is that represented on your demonstrative slide 5,
3 in concept?

4 A. Okay. So source code, such as illustrated on that easel,
5 is compiled using the Java programming language compiler, and
6 that creates this executable bytecode.

7 Executable bytecode, which is what the virtual
8 machine computer actually executes, are then stored in the
9 Class library.

10 Q. What are we showing on the next slide, slide 6?

11 THE COURT: Would this be a good point to take a
12 break?

13 MR. JACOBS: Yes, Your Honor. Perfect. Thank you.

14 THE COURT: All right. Remember the admonition.
15 We'll see you back here in 15 minutes. Thank you.

16 THE CLERK: All rise.

17 (Jury out at 9:30 a.m.)

18 THE COURT: Please, be seated. Are there any issues
19 the lawyers wish to take up with the Court?

20 MR. VAN NEST: Yes, Your Honor.

21 THE COURT: Please, go ahead.

22 MR. VAN NEST: Just in the nature of a heads up and
23 to get your guidance on how to deal with it, I think following
24 Dr. Mitchell the plaintiff plans to call Mr. Rubin on cross.
25 And I think, according to Mr. Jacobs, they'll complete their

1 examination of Mr. Rubin today.

2 Now, they also gave us notice that they want to call
3 Mr. Schmidt. He's not available today, but he's available
4 tomorrow. And we've agreed that he will be examined by them
5 tomorrow.

6 My expectation or my game plan suggestion was, once
7 they finish whatever examination they're doing of Mr. Rubin,
8 I'd like to simply call Mr. Rubin in my case, because he's a
9 fairly central witness, he's here, he's on the stand.

10 So I was going to suggest that when they finish their
11 exam, the Court would indicate whatever you wanted to indicate
12 that now Google is going to get its opportunity, present its
13 case, and I'll start with Mr. Rubin while he's here. And.

14 Then Mr. Schmidt -- we may need to interrupt
15 Mr. Rubin tomorrow, to get Mr. Schmidt on, but we both agreed
16 that when Schmidt comes, you know, they can reopen their case
17 to handle Mr. Schmidt.

18 **THE COURT:** Is Mr. Rubin your last witness?

19 **MR. BOIES:** Yes, Your Honor.

20 **THE COURT:** So what do you say to this proposal?

21 **MR. BOIES:** This is agreeable to us. In fact, we've
22 talked about it.

23 Your Honor, we've talked about it, and this is
24 perfectly agreeable, and, we think, a sensible way to approach
25 it.

1 **THE COURT:** All right.

2 **MR. BOIES:** I just want to give you a heads up
3 because what we'll ask for is I'll stand up when Mr. Rubin's
4 cross is done and say, "Your Honor, I'll reserve my examination
5 for our case which is just about to start," so that the jury
6 doesn't get the idea that I'm not going to examine Mr. Rubin at
7 all. That's all.

8 **THE COURT:** But you are -- in other words, the --
9 well, when does the plaintiff stand up and say, "We rest"?

10 **MR. VAN NEST:** I guess they would do that after their
11 examination of Mr. Rubin, would be the logical thing. And
12 that's fine with me.

13 **MR. BOIES:** I think we could do it either one of two
14 ways, Your Honor. Either we could say, at the end of
15 Mr. Rubin, "We rest subject to calling Mr. Schmidt tomorrow."
16 Or we could wait until Mr. Schmidt is off and say, at that
17 point, we rest. We could do it either way.

18 **MR. VAN NEST:** I would prefer -- when I'm examining
19 Mr. Rubin, I kind of wanted to go back and start at the
20 beginning, and so on and so forth. So I would prefer that to
21 be in my case.

22 I would prefer the former of what Mr. Boies
23 suggested: We rest subject to calling Mr. Schmidt.

24 **MR. BOIES:** And that's agreeable to us, if it's
25 agreeable to the Court.

1 **THE COURT:** We'll do it that way. Then it will
2 brighten the heart of the jury to hear someone say they rest.

3 (Laughter)

4 **MR. VAN NEST:** I was thinking that myself.

5 **THE COURT:** And that -- even if it's subject, too.
6 So that's the way we'll do it. And we will -- for technical
7 purposes, it will be deemed as if any Rule 50 motion is
8 reserved for later.

9 Everybody agree to that?

10 And this would be even though it would have to
11 include Mr. Schmidt's testimony.

12 **MR. VAN NEST:** Yes, Your Honor.

13 **MR. BOIES:** Yes, Your Honor.

14 **THE COURT:** That's all fine. See you back here in 15
15 minutes. Thank you.

16 (Recess taken from 9:34 to 9:51 a.m.)

17 **THE COURT:** Let's go back to work. All set and ready
18 to go?

19 **MR. JACOBS:** Yes, Your Honor.

20 **THE COURT:** I'll bring in the jury.

21 (Pause.)

22 **THE COURT:** All right. Here's another note from
23 somebody on the jury, one:

24 "Can you explain the difference between a
25 package, for example, java.nio, and an API?"

1 And then here's another question:

2 "For later, how does copyright process work?

3 What can/cannot be copyrighted, and is this

4 predetermined when copyright is granted?

5 Does it expire?"

6 All good questions. All right. Thank you.

7 **MR. JACOBS:** Your Honor, would it be okay with you if
8 I simply asked that first question --

9 **THE COURT:** The first question you can ask, yes. Go
10 ahead. Let's bring in the jury.

11 (Jury enters at 9:53 a.m.)

12 **THE COURT:** Welcome back. Be seated.

13 And we'll continue right on. Let's make sure the
14 jury has got their notepads out and ready.

15 **BY MR. JACOBS:**

16 **Q.** Dr. Mitchell, what are we showing on slide 6 of your
17 slides?

18 **A.** Slide 6 has two sides. On the left side is a screen shot
19 of a browser view of the Oracle Java API. And on the
20 right-hand side is the same illustration from the previous side
21 showing a schematic of the Java Class Library with executable
22 bytecode.

23 And the arrows from left to right will show the
24 correspondence between the API as we see it in the Web browser
25 when you look at it and the executable class library.

1 Q. So let's first understand where we are in the hierarchy.
2 What are we looking at when we're looking at java.nio.Channels
3 Class Channels?

4 A. Okay. The notation on the left, in the browser view,
5 tells you that you're looking at a class.

6 Q. And that's the highlighting of class right in the top
7 there?

8 A. That's one way to look at it. I was looking a little bit
9 lower. The biggest print on this display is the word "Class"
10 and the word "Channels." So that's telling you in big letters
11 that you're looking at a class, and the class is named
12 Channels.

13 The smaller type above that says where this class is.
14 And it's in the java.nio package Channels portion of that.

15 MR. JACOBS: Dr. Lee, can you highlight java dot --
16 yes. No, down. And highlight java.nio.Channels Class
17 Channels, the biggest lettering on the top. Okay. That's what
18 we're looking at.

19 THE WITNESS: Another indication of how this class is
20 related to other classes is the little diagram just below the
21 big class Channels type. And it says that java.lang object is
22 a top-level class. And then below that,
23 java.nio.Channels.Channels is a subclass of that.

24 So in the class hierarchy, which is different from
25 the package organization, this class is a subclass of the

1 object class, which is at the top of the class, which is at the
2 top of the class hierarchy. Every class is a subclass of
3 object, either directly or indirectly.

4 **BY MR. JACOBS:**

5 **Q.** So what else are we illustrating on this slide?

6 **MR. JACOBS:** Let's go back to the base image,
7 Mr. Lee.

8 (Document displayed.)

9 **THE WITNESS:** Below this top part there's a list of
10 summary of methods from the class. And you can see that they
11 correspond to the methods in the class library. So the API is
12 telling you how each of the methods, the operations on the
13 class's objects are declared.

14 And that corresponds, that's information about the
15 actual executable class library so that programmers can use the
16 prewritten programs in the class library through these method
17 declarations and the names that they tell the programmer.

18 **BY MR. JACOBS:**

19 **Q.** So let's look at the next slide. And what are we showing
20 there?

21 (Document displayed.)

22 **A.** Below the method summary in the browser view, there's
23 detail about each of the methods. So this is really -- this
24 second slide is a screen shot of what you see if you scroll
25 down on the Web browser on the left.

1 And there are two methods in this screen shot of the
2 browser view, and they point to the compiled methods, the
3 executable code that programmers use in order to operate on
4 objects of this type.

5 Q. So let's just understand the hierarchy. What package are
6 we in?

7 A. We're in java.nio.Channels.

8 Q. And what class are we in?

9 A. The Channels class that's in that package.

10 Q. And what are we looking at on slide 7?

11 A. These are methods that are in the Channels class of
12 java.nio.Channels.

13 Q. And all of this would be seen if you were to use your
14 browser to scroll through the entire description of that class;
15 is that right?

16 A. That's correct. The browser lets you navigate up and down
17 in the class and package hierarchy. And when you reach a
18 specific class, you can scroll down to see more information
19 about the class. Typically, it's several screens long to tell
20 you all the information in the API about a class.

21 Q. So what are we seeing on slide 8?

22 A. Slide 8 is information -- is a schematic showing how the
23 two parts of the previous slide are produced. Programmers and
24 architects write source code, and the source code is stored in
25 source code files. So that's where the Java programming

1 language is used to define programs in the library and other
2 kinds of programs that use the library.

3 Source code is then used to produce two kinds of
4 things. One is the API documentation, the thing you can see on
5 the Web if you're looking at the API. And the other is the
6 executable code in the library so that programmers can use the
7 things defined in the source code.

8 So there are two different programs, these things
9 with gears in them are programs that convert the source code
10 into two different forms. One thing with gears produces this
11 nice viewable API description that you can look at on the web.
12 And the other thing with gears, the language compiler,
13 translate the source code into executable bytecode.

14 One of the things that follows from the way that this
15 is arranged is that the correspondence that we saw in the last
16 few slides between what you see on the web and the executable
17 library is not an accident. It's a result of the way those two
18 things are produced from source code files.

19 **Q.** So how does the structure of the code in the Java Class
20 Library match the structure of the documentation?

21 **A.** All of this arrangement of classes, their fields and
22 methods, the relationships between classes, the subclass
23 relationship, the relationship between classes and interfaces,
24 which classes support each interface, the arrangement of those
25 elements into packages, and the hierarchy of packages and

1 subpackages are all reflected in the class library because
2 they're produced from the same source code.

3 Q. Is this true -- to what extent is this true across the 37
4 API packages that are in dispute in this lawsuit?

5 A. This illustration showing the way that the source code is
6 used to produce the API on the Web and executable class library
7 files applies to all packages, including all the ones being
8 discussed in this case.

9 Q. What is the relationship between a package and application
10 programming interfaces?

11 A. A package is a grouping within a library. API can refer
12 to the API of a full library or it could talk about the API of
13 a single package or the API of the class. Generally, we're
14 using API to refer to the interapplication program interface,
15 all of these characteristics of a group of packages.

16 Q. So to go back to the poster, the -- on the easel, Trial
17 Exhibit 3452, what is the relation -- where is the API on this
18 poster and how does that relate to a package?

19 A. The -- there are portions of an API on this poster.
20 There's the declaration, which is included in an API. API also
21 has relationships between different classes and interfaces and
22 packages. There's some picture of this partially represented
23 on the easel.

24 API also includes a description of how these things
25 work, what the names mean, how the methods are used. And

1 that's not shown on the easel.

2 Q. How would this particular API relate to a package? We see
3 up at the top it says "Package java.lang," et cetera. Do you
4 see that?

5 A. Yes.

6 Q. What's the relationship between package and this
7 particular application programming interface described here?

8 A. As far as I understand what's written, that's indicating
9 that this class declaration is -- should be considered a
10 portion of this package, an element of the package.

11 Q. Do you consider API design to be a rote or a creative
12 activity or something else?

13 A. API design is really a creative process. There are many,
14 many kinds of decisions that go into API design. Often, the
15 people who do API design are called software architects. They
16 arrange the classes.

17 So to begin with, if you're designing a library and
18 it's API, you'd start by thinking about what kinds of problems
19 you're going to help programmers solve. Select some classes
20 that would provide objects in object-oriented programming to
21 help programmers solve those problems.

22 For each class, an API designer selects methods, ways
23 of operating on those objects, that would help programmers
24 solve problems.

25 Once the classes are identified, they're arranged in

1 a subclass hierarchy to make it easier to write the
2 implementation and to make it easier for programmers who use
3 the library to understand how the different concepts in the
4 library work together.

5 Interfaces are used to make it possible to write code
6 that operates over many different classes in the same way
7 showing commonality across those classes.

8 And then the organization into packages is helpful in
9 informing programmers how to understand where each solution,
10 each program they might want to use in buildings their system,
11 sits in the library. And to make a summary and conceptual
12 organization clear across the library.

13 **Q.** And how are those choices illustrated on the slide 9 we
14 saw before and is up again?

15 **A.** If -- if you remember, there are -- each class has
16 methods. So this shows an example of the methods in one class.
17 The hierarchy of Buffer, ByteBuffer, MappedByteBuffer and how
18 these classes are related to each other is shown in this slide.

19 And then it's a little bit gray in the slide, but
20 this package -- these classes shown here are in the nio
21 package. That package is part of a larger library of
22 interrelated packages with classes that refer to each other in
23 various ways.

24 I don't think I mentioned yet that each method in a
25 class has parameters, data that must be supplied to the method

1 in order for it to do its job, and returns a value of a given
2 type.

3 The types of the parameters and the return type of
4 the method can be classes from anywhere in the library. So
5 that's another way that code in one portion of a library, in
6 one package, can depend on and use code in another.

7 The arguments to a method, the parameters to the
8 method and the return can be classes from other packages.

9 **Q.** How does designing an application programming interface
10 compare with other aspects of writing software?

11 **A.** When you start to design an API for a portion of a
12 library, you really start with a clean slate. So at that point
13 nothing has really been determined. The high-level decisions
14 that govern the organization of the whole software system can
15 start with the API. So that's the starting point.

16 And then from there, as more and more decisions are
17 made, the remaining decisions are more constrained. That's not
18 to say there isn't creativity involved at all levels, but there
19 certainly is quite a bit at the beginning in order to map out
20 the organization of a system.

21 **Q.** Let's take a look at a -- an example of an API concept as
22 architected in different contexts.

23 Can you explain to us what's shown on slide 10?

24 **A.** Certainly. This is one of my favorite examples. I
25 usually teach the -- some elements or aspects of the Smalltalk

1 collection hierarchy in my class every year because it's a
2 portion of the Smalltalk library that involved over a period of
3 time and involved a lot of intellectual effort.

4 The Java design also reflects a number of years of
5 work and refinement. A collection in programming is an
6 arrangement of data used by a program to store and retrieve
7 data in certain ways.

8 One kind of collection that appears in each of these
9 is the idea of a list. A list is just data elements in order.
10 And in the Java hierarchy you can see that list is directly
11 below collection.

12 In Smalltalk there's some other concepts in between.
13 The immediate subclass of collection is called sequenceable
14 collection. And then ordered collection is a special case in
15 the Smalltalk design of sequence.

16 So the designers of the Smalltalk library thought
17 this intermediate concept was useful in organizing the library
18 and helping programmers use it. In the Java design, that it's
19 simpler, and List is just immediately a subclass or
20 subinterface of Collection.

21 **Q.** So you were looking at the green in slide 10?

22 **A.** Yeah. The way this is laid out, the colors are used to
23 show how the same basic concept appears in two or more
24 examples.

25 These examples are written in different languages,

1 but I think the main difference, and the reason why we have
2 different designs, is just different teams at different times
3 approach the problem differently.

4 Q. So just to start at the beginning, Java is obviously a
5 programming language. What is Smalltalk?

6 A. Yeah, sorry. Java is obviously a popular programming
7 language. Smalltalk is an earlier object-oriented language
8 that was very popular for a few decades. I think you still
9 find some people using it, but it's not as popular these days.

10 And C++ is another major currently popular
11 object-oriented language.

12 Q. And associated with each of these programming languages
13 are their API libraries?

14 A. Yes, there are extensive libraries for all of these
15 languages.

16 Q. And is there a library in each -- in Smalltalk and C++
17 that does roughly what the collections hierarchy does in Java?

18 A. Yeah. In each of these, they are a portion of a larger
19 library. And it's often referred to as the collections classes
20 or the collections library of that class.

21 Q. And if you were to line them up and overlay one on top of
22 the other, would you see lots of similarities or differences in
23 the way the APIs are designed?

24 A. You see some differences. And you can imagine that by
25 trying to line up these three pictures on top of each other.

1 They are different height. They are different width. And
2 there are different degrees of differentiation in the three
3 designs.

4 Q. Now, does Java itself illustrate anything about the
5 flexibility available to -- let me start over again.

6 I've got package java.util on the screen. Do you see
7 that?

8 A. Yes.

9 Q. And in the next slide I've got java.util from Java SE 5.0.
10 Do you see that? It's on the screen.

11 A. Yes.

12 Q. And what do these two slides illustrate?

13 A. These are two pictures of the java.util package at
14 different points in time. So earlier in the development of
15 Java, Java 1.0, in the '90s, there were -- there was a much
16 smaller library. A team had spent less time developing this.
17 It was earlier in the history of Java. It was just a smaller
18 library and there were fewer concepts in it.

19 And over time -- because this is a very important
20 part of the library, collections and related things are widely
21 used -- the process led to a much more detailed and elaborate
22 design with more concepts and more interrelationships between
23 them.

24 Q. So now we've got Java 1.0 and 5.0 on the screen together.
25 Can you briefly tell us what this is illustrating?

1 **A.** This is a side-by-side comparison of the Java.util package
2 at different points in time, different releases of the Java
3 system. The highlighting shows the original classes and where
4 they appear now.

5 You can see that some are -- some have new classes
6 between them, and object. They are arranged in the hierarchy
7 in a different way. And they are spread out in the picture
8 illustrating a number of other classes that have been added.

9 On the right column, you can also see that the set of
10 interfaces associated with this package has become much richer,
11 supporting other forms of reuse or code using different classes
12 in a uniform way.

13 So the picture has gotten much more complicated.
14 That reflects more sophistication in the design, in the
15 collection of entities, and their interrelationships and
16 structure.

17 **Q.** I'd like to now turn to the subject of copying,
18 Dr. Mitchell, and I'd like to ask you if you conducted any
19 investigation into the copying by the developers of Android
20 from Sun's or now Oracle's Java APIs and code?

21 **A.** Yes, I did. I did a number of things to look into this
22 myself. I looked at three categories. One has to do with the
23 correspondence between the APIs and how the Android API is
24 copied from the Oracle API.

25 Another has to do with the way the API is used in

1 code, the way that the Android code copies the Oracle API into
2 the code. And, third, the way that other code sources were
3 used to copy into the Android library.

4 **Q.** Let's take that first category, which is
5 specification-to-specification or documentation-to-
6 documentation copying. How did you investigate this -- this
7 topic?

8 **A.** Well, to get a feel for this myself, I spent a lot of time
9 periodically, and over the time I've been involved in this case
10 over and over again, looking and comparing different sections
11 of the API to see how they compare.

12 Because the APIs in both cases are available on the
13 web, it's fairly straightforward to open two web browsers side
14 by side and navigate through the two APIs and libraries
15 simultaneously in the same way and compare the way one looks on
16 the screen with the other.

17 By and large, that -- they are really identical.
18 They are displayed in different colors and so on. But the
19 content is the same. You see the same classes in the same
20 hierarchy in the same -- in packages of the same name
21 supporting the same interfaces and arranged in -- in the same
22 way.

23 There are a few minor differences. Every now and
24 then there's a method in the Oracle API that's not copied into
25 Android. But, basically, looking across the 37 packages, they

1 are really, essentially, identical.

2 There are a couple of packages that have a few more
3 differences than that, but those are the exception.

4 **Q.** So let's look at an example I put up on the screen: Class
5 IntBuffer. What are we looking at on this slide 14?

6 **A.** This is an illustration to try to make the main points
7 visible. The left part of this is another screen shot of what
8 you see in a web browser when you look at the Oracle Java API.

9 And then, because the browser view takes a few
10 screens to see all of the important parts, there's a blowup on
11 the right that shows something from the top, the class and its
12 arrangement in the hierarchy in relation to interfaces, and
13 then something from the middle of the browser page, the method
14 summary.

15 **Q.** So what are we seeing now, Dr. Mitchell?

16 **A.** This is basically the same view of the Android library API
17 displayed in the same way.

18 On the left, so that's the lower left of these four
19 boxes on the screen, is the screen shot of how this looks in
20 the browser. And you can see that the basic layout and the
21 tool used to display these things uses the same idea and shows
22 things in the same way.

23 And then the blowup to show the key elements from the
24 browser view makes it a little bit easier to compare the two
25 since things are highlighted and shown in a bigger way.

1 Q. And what are you able to deduce by comparing these side by
2 side in this way, with the Oracle documentation being 610.2 and
3 the Android documentation being Trial Exhibit 767?

4 A. Basically, you can see that you have the same class in the
5 same library with the same methods. Everything is basically
6 the same. You start at the top. You have the class IntBuffer
7 shown in both places. The slightly pink highlighted area shows
8 where this class hits in the subclass hierarchy in relation to
9 other classes.

10 In both cases, this IntBuffer class is a subclass of
11 java.nio.Buffer class. In both cases the java.nio.Buffer class
12 is a subclass of Object. So, really, same picture, same
13 information, same meaning, same API.

14 Similarly, if we look at the relationship between
15 classes and interfaces, in both cases this class implements
16 Comparable. "Comparable" in both libraries is in the java.lang
17 package and declared in the same way.

18 And, similarly, if we look down at the methods, we
19 see -- basically, we see the same method declarations with
20 basically the same information and explanation of what those
21 method declarations mean.

22 Q. And let's focus on that last part, the explanation of what
23 they mean. What's the significance of that similarity?

24 A. The explanation is important because it says how each
25 method works, what it does, what it accomplishes. And the

1 explanation as used by programmers who want to solve a
2 programming problem, they're looking for methods that will help
3 them solve their problem.

4 And the explanation is also used by implementers.
5 When an API is designed and presented to coders, they can use
6 the explanation to write code to implement the narrative or
7 explanation given in English language.

8 **Q.** Now, again, back to our hierarchy, this is a class within
9 a package?

10 **A.** Yes. This class is in the java.nio package.

11 **Q.** And, roughly, how many classes are in the 37 packages that
12 are in dispute?

13 **A.** In 37 packages, there are around 400 classes.

14 **Q.** And then in terms of the elements that you analyzed below
15 the class level, what are those elements of?

16 **A.** There are a number of other things, but the main ones are
17 the methods and fields. There are probably 5,000 methods
18 across these. So that's a very large amount of information
19 that's present in the Java -- Oracle Java API, and it's copied
20 into Android.

21 There's a -- there are some other things I haven't
22 mentioned, the exceptions, enums, other elements --

23 (Reporter interrupts.)

24 **A.** Enums, e-n-u-m-s.

25 All of these elements are copied and sit in the same

1 organization in both of these libraries.

2 Q. And, roughly, what percentage of copying did you observe
3 in your work?

4 A. Quantitatively, approximately 90 percent. There's --
5 occasionally there's something that's not copied, and
6 occasionally there may be some small differences in the way
7 things are named.

8 For example, the methods have parameters. Sometimes
9 the parameter names are different. Surprisingly, they're the
10 same in probably two-thirds of the cases, even though that's
11 not necessary for the library to be used in the same way by
12 programmers.

13 So I'd say both of those facts, that the -- that the
14 names and the structure corresponds, you know, almost uniformly
15 throughout and that there are additional features that really
16 wouldn't need to be copied just in order to provide the same
17 function to programmers are also copied.

18 Q. Now, so what is your overall opinion with respect to
19 whether the APIs for these 37 packages were copied by Google
20 into the Android APIs?

21 A. I don't think there's any way that a separate team could
22 have come up with so many things that are identical except by
23 copying the original API. There are just thousands of things
24 that match and I don't see how that could have happened in any
25 other way than by copying.

1 Q. Now, your second category was copying from the API
2 documentation or specification into the source code. Did you
3 study that relationship?

4 A. Yes, I did. One way I looked at that is by using the
5 browser to go through the API and comparing that to the source
6 code that I had.

7 In addition, we saw the method by which the browser
8 view was created. And you can see that that -- since it's
9 derived from the source code, it's going to match throughout.

10 Q. So let's take a look at a comparison of Trial Exhibit
11 610.2 and 46.22, an extract of which is on the screen.

12 Can you tell us what this is showing us?

13 A. This is a view with two sides. The left side shows the
14 Oracle Java API, as displayed in the way we've been looking at
15 it, and the right side shows Google Android code.

16 Q. And if you follow along on your screen, we'll highlight
17 various portions of this.

18 What are we looking at right now?

19 A. Start it at the largest collection or -- in the
20 organization, "Packages." And this -- on the left is
21 describing some portion of the java.nio package. In the code
22 there's a declaration saying this is the java.nio package or
23 this is to be included in the java.nio package.

24 Q. And then I've highlighted it more in the middle of the
25 pages shown on the screen in demonstrative slide 19.

1 **A.** So the API contains declarations. The declaration shown
2 in the middle of this screen capture is a declaration for a
3 class, the class IntBuffer. And that declaration is copied
4 directly into Google code to give the Google package and class
5 arrangement the same structure as the Oracle API.

6 This declaration says that IntBuffer is a subclass of
7 Buffer. And it also says that IntBuffer implements the
8 comparable interface.

9 So this declaration gives the relationship between a
10 class, another class, and an interface. And it's the
11 collection of these declarations that give the class library
12 the hierarchal and complex structure that we saw in the poster
13 and other works.

14 **Q.** And then let's go down one more level and look at
15 "IntCapacity."

16 **A.** This is the allocate method that's one of the methods of
17 the IntBuffer class. And every element that's presented in the
18 library to be used by programmers has to be declared in some
19 way. So there are declarations giving the package names.
20 There are declarations for classes. And then since a class is
21 a collection of methods and fields, there are declarations for
22 each method and field.

23 This method declaration says this is a -- because of
24 where it's located, a method of the IntBuffer class. And the
25 parameter, the data that needs to be supplied to the method, is

1 called "Capacity." That has to be an integer. And when
2 "Allocate" is called, it returns an IntBuffer, the third term
3 here.

4 Q. So looking at this example, in sum, what are we seeing
5 that is present in the API specification and copied into the
6 Android code?

7 A. The code -- the declarations, which are really in the Java
8 language syntax, they are code in the API, are copied into the
9 source code for the Oracle library -- I'm sorry, for the Google
10 library.

11 Q. Let's look at another example, looking at slide 20. And
12 what is slide 20 showing us or going to show us?

13 A. Slide -- the next slide has a little more detail about the
14 methods. Each method in the API has an English description of
15 what that method does, how it works.

16 There's a narrative explanation of the meaning of the
17 element declared in that portion of the API. And that's
18 reflected in the source code because the source code was
19 written to achieve the purpose that is described in the API.

20 Q. So let's start with protection domain on the left and the
21 corresponding code on the right. What are we illustrating
22 here?

23 A. "ProtectionDomain" is what's called a constructor. It's a
24 special kind of method that creates new objects. And so
25 there's a declaration syntax. And ProtectionDomain requires

1 two parameters. And then an English language description
2 saying what that means, what it does, how the constructor
3 operates.

4 And you can see in the source code, if you're a
5 source code reader, that the source code does exactly what the
6 English sentence calls for. Namely, the source code creates a
7 new protection domain, and it gives it the code source and
8 permissions that are passed as parameters.

9 Q. So let's start with the relationship between -- from
10 declaration to declaration. What is illustrated there about
11 that?

12 A. The declaration is the same, so that's copied from the API
13 into the code.

14 Q. And then the narrative is reflected how in the source
15 code?

16 A. Uhm, the narrative is reflected in the source code because
17 the source code is a program that in a sense carries out that
18 narrative, does what the explanation requires for this method.

19 Q. If one were to look across the 37 packages, as you did,
20 what would one see then in terms of the kinds of similarities
21 you've shown in this example?

22 A. Uhm, this is representative. The API explains how the
23 elements behave and the code in a sense makes good on that
24 promise, that the API wouldn't be useful to programmers if it
25 wasn't an accurate description of the way the code behaves.

1 Q. And I'm going to move us along then to the next category
2 of copying that you analyzed, Dr. Mitchell. And that's the
3 code-to-code copying, correct?

4 A. That's correct.

5 Q. And what do we mean by that?

6 A. These are other examples of code out beyond the API that
7 are copied from the Java code base into -- the Oracle Java code
8 base into the Google Android code base.

9 Q. So this is rangeCheck, which the jury saw with Mr. Bloch,
10 and we have 623.1 on the left and 46.7 on the right. Did you
11 analyze this?

12 A. Yes, this is one of two categories of code copying into
13 this kind. I looked at this quite a bit, to try and understand
14 what this method does, how it works, and how similar it is in
15 both cases.

16 Q. Now, on the Java side it comes from a file called
17 java.util.arrays.java, correct?

18 A. That's correct.

19 Q. And on the Android side, where can this rangeCheck code be
20 found?

21 A. I think this code is found in two files. One of them is
22 the Timsort.java. And the other is another Java file with a
23 similar name, ComparableTimSort.java.

24 Q. How does this code, if at all, relate to any of the API
25 packages that are at issue in this case?

1 A. Well, this is in -- you can read from the name,
2 java.util.arrays. It's from the java.util package.

3 Q. What conclusions did you reach about the similarity of
4 rangeCheck in Java and Android?

5 A. Well, these are really strikingly similar. They're really
6 identical up to details that don't really matter as far as
7 causing the method to work properly.

8 One of the things that I just found really unusual
9 and unexpected is the spacing around the plus signs, which
10 seems kinds of arbitrary. You could type the spaces as you
11 like and there are some places where the spaces are -- where
12 there's spacing around the plus and some not. It just seems
13 unlikely that anybody would do that twice by accident. Really
14 looks like -- I don't know how this could happen except by
15 copying the code.

16 Q. How does this relate to Google's claim that it conducted a
17 clean room implementation of the core libraries?

18 A. Since I don't see how this could happen without copying
19 the code, it's clear that whoever inserted this Google code
20 into that code base had access to the code that's in the Oracle
21 Java code base.

22 Q. Now, does this code actually find its way onto mobile
23 telephones?

24 A. I believe this does. As a part of the library, this is
25 compiled and then can operate on a mobile device or phone.

1 Q. Did you do any investigation to test that proposition?

2 A. Yes, I did. I actually instrumented a -- the code that
3 runs on the phone and installed it and ran it on the emulator
4 to see that this was used in running the phone.

5 Q. And what phone were you analyzing?

6 MR. VAN NEST: Objection, Your Honor. Outside of the
7 report.

8 THE COURT: Is this in the report?

9 MR. JACOBS: Paragraph 221.

10 THE COURT: This is Exhibit 67?

11 MR. JACOBS: Yes, this is going to simple answer to
12 the name of the vendor and phone that's referred to.

13 THE COURT: Paragraph 221 refers to readme text file.
14 That's not the same one we're talking about, is it?

15 MR. JACOBS: Your Honor, withdraw the question.

16 THE COURT: Thank you.

17 MR. VAN NEST: Your Honor, could the portion of
18 Dr. Mitchell's prior answer be stricken? I believe it's on the
19 same issue.

20 THE COURT: I don't -- I'm not going to strike
21 anything that's already been said. But we will stop the
22 inquiry at this point on that limited subject.

23 All right. Go ahead.

24 BY MR. JACOBS:

25 Q. Now, did you conduct analysis of another kind of copying?

1 A. Yes. There are two other cases. One has to do with
2 comments, and the other has to do with decompiled code.

3 Q. So let's talk, first, about decompiled code.

4 Have you prepared some slides to illustrate what
5 decompilation is?

6 A. Yes, I have. Developers write source code in the Java
7 language. And then in order to produce executable programs
8 that are run, the source code is compiled using a compiler into
9 a class file.

10 It turns out that there's a program called a
11 decompiler that can do a form of the reverse. A decompiler
12 takes compiled class files and produces Java source code files
13 from the compiled class file.

14 Q. So in a nutshell, what is decompilation?

15 A. Decompilation is undoing the work of the compiler to get
16 the source code back from the class file.

17 Q. And why would somebody who wanted to copy code go through
18 this exercise?

19 A. It's a quick and easy way to get source code if you need
20 it, if you have the class files at your disposal.

21 Q. So the starting point in the decompilation examples that
22 you -- that you investigated was what?

23 What was the original material that was put through
24 the decompiler?

25 A. Well, the Oracle source code has files for a number of

1 different classes. Those are compiled into class files. Any
2 computer that runs Java will have the class files on it so
3 those are accessible to programmers and engineers using Java.

4 Apparently, one of those class files that are widely
5 distributed was used to produce the source code that's in the
6 Google library.

7 **Q.** So let's look at that. I've got slide 24 from Trial
8 Exhibit 1031 on the screen. What are we looking at here?

9 **A.** Uhm, this is the Android source code. And it's not in the
10 form that a person -- a normal human programmer would write.

11 So you can see from looking at this, if you're a
12 programmer or familiar with code, that this appears to be
13 automatically generated.

14 **Q.** And when you say "automatically generated," what do you
15 mean? And what are you pointing to in this exhibit?

16 **A.** The declaration of one of the methods has a number of
17 parameters. And programmers usually give the parameter name --
18 parameter some meaningful name so they can remember what they
19 are and how they're to be used.

20 A decompiler working from a class file doesn't have
21 any way to understand what the code actually does. It just has
22 to automatically make up names in any way that it can, to make
23 the source code correctly formed.

24 **Q.** So we're looking at PolicyNodeImpl -- i-m-p-l -- .java in
25 Android. And just looking at the code itself, what were you

1 able to deduce?

2 **A.** Well, this method has two parameters that are sets and two
3 parameters that are booleans. A boolean is either true or
4 false. It's a flag saying whether something should be done or
5 not.

6 And the decompiler -- this particular decompiler just
7 makes up names for parameters based on their type. So it
8 doesn't know what kind of set is being used here, it just calls
9 the parameter set because it doesn't know any better. And it
10 needs to do that twice so it just adds a number to the second
11 one to make it different from the first.

12 I think these are actually some kind of policy sets
13 and qualifier sets. In the original source code, the
14 decompiler doesn't know anything about those, so it just makes
15 up some random-seeming names instead.

16 **Q.** Now, did you do some further investigation to test whether
17 your -- whether these indications of decompilation from Oracle
18 bytecode into Android's source code were correct?

19 **A.** The next slide shows side-by-side two source code files.
20 The one on the left is the Android source code file that we
21 have been looking at, and the one on the right is produced in a
22 similar way to make sure that this assessment is correct.

23 The source code on the left is produced by taking the
24 Oracle class library and running it through the decompiler to
25 see if that produces something that matches the Android code.

1 There is a difference at the top in that someone
2 pasted in a comment at the top that's some form of -- looks
3 like a license of some form, but below that the actual code
4 matches completely.

5 So I think that's strong evidence that the Google
6 code was actually produced from the Oracle class file by a
7 particular decompiler called JAD for Java Decompiler.

8 **Q.** Now, you mentioned some other files in your analysis.
9 Let's just get those into the record and then I'll ask you a
10 question about them.

11 AclEntryImpl, AclImpl, GroupImpl, PermissionImpl,
12 PrincipalImpl, and AclEnumeratorImpl. And when I say
13 "mentioned," you mentioned in your report.

14 What is your analysis as to those files?

15 **A.** It's basically the same story. The Android code is
16 decompiled class code and the correspondence, when you check it
17 by doing the decompilation ourselves, is exactly the same as
18 the slide here.

19 **Q.** So what does this tell you about whether the Android code
20 was developed in a clean room?

21 **A.** This Android code is produced by running a decompiler on a
22 copied class file.

23 **Q.** A class file from?

24 **A.** From Oracle.

25 **Q.** And, therefore, what does that say about the clean room?

1 A. The developers had access and used the Oracle installation
2 and copied from it, in this case by using a decompiler to
3 produce source code by this quick and easy method.

4 Q. Now, I've highlighted -- just to close this illustration
5 off, I've highlighted the same variables on the left and the
6 right.

7 Can you just tell us what that is signalling?

8 A. There's is just showing that the parameter names in both
9 decompiled codes are the same. So that's just showing that
10 this really is the output of a decompiler by this
11 characteristic, this kind of one of the telltale signs that
12 this is automatically generated machine-produced code rather
13 than written by a programmer independently.

14 Q. And then I'm -- on Slide 26 I've got OwnerImpl.java from
15 both Java and Android on the screen. What is this showing us?

16 A. This is another example from another one of these files
17 and it's just showing, again, the parameter names are produced
18 automatically. The parameter name is just the name of its type
19 with a number added. It's kind of like -- well, it's using an
20 automatic method to choose parameter names that are not really
21 meaningful. They are just numbered.

22 Q. And what is that telling us them about how OwnerImpl.java
23 in Android was created?

24 A. I don't see that there -- I can't think of any other
25 explanation other than someone took a class file, an Oracle

1 class file, and ran a decompiler to produce this source code.

2 Q. Now, did you look at any other kinds of copying? You
3 mentioned a third category.

4 A. There's some literal copying of comments, and that's
5 another way of understanding that there really wasn't a clean
6 room situation. The comments in some of the source code,
7 admittedly a small portion, but enough for us to see that the
8 Android developers had access and used things that were
9 contained in the Oracle code.

10 Q. So let's just walk the jury through a couple of lines of
11 what you identified as copied code.

12 (Document displayed)

13 Q. On the left we have got CodeSource.java in the Java code,
14 and on the right we have CodeSourceTest.java in Android,
15 correct?

16 A. That's correct.

17 Q. Just give us a couple of these lines and walk through them
18 side-by-side?

19 A. I could just read the first one if you want. It says:

20 "If this object's port (get location)" -- and
21 then some code saying how the port is
22 identified -- "is not equal to minus one,
23 then it must equal some other value."

24 And that's exactly the same thing with the same
25 syntax for the name -- referring to the port, same

1 parenthesization. It's saying the same thing.

2 The thing that's taken out are some html commands,
3 but otherwise the English sentence and what it says is
4 syntactically, you know, identical. It's the same sentence.

5 **Q.** Do you look at something called CollectionCertStore
6 parameters?

7 **A.** There were two files and I think that's the other file
8 that has comment code.

9 **Q.** And what did you find in those files?

10 **A.** The same kind of comment copying.

11 **Q.** I want to go back to rangeCheck. And with reference to
12 Paragraph 240 of your report, Dr. Mitchell, is the line-by-line
13 copying of rangeCheck code actually found on a particular phone
14 that you analyzed?

15 **A.** This is on a Samsung phone, I believe.

16 **Q.** And did you check that?

17 **A.** Yes, I did.

18 **Q.** How did you do that?

19 **MR. VAN NEST:** Objection, your Honor. I don't see it
20 in the report.

21 **THE COURT:** Is it in the report?

22 **MR. JACOBS:** It is, your Honor.

23 **THE COURT:** Where is this in the report?

24 **MR. JACOBS:** Paragraph 240.

25 **THE COURT:** What do you say to that, Mr. Van Nest?

1 **MR. VAN NEST:** I'm saying that this reflects looking
2 at the source code, which he's already talked about, not
3 looking at the phone.

4 He's already testified about looking at source code,
5 not looking at a phone. That's not reflected here.

6 **MR. JACOBS:** It finishes with, the last sentence of
7 that, your Honor.

8 **THE COURT:** All right. It's a close call, but it's
9 close enough so the Court will allow the testimony.

10 Objection overruled. Go ahead.

11 **A.** Basically the story here is that the Google code
12 distribution is used by the phone vendors, and on their sites
13 you can find instructions about how developers should use the
14 code that they provide --

15 **MR. VAN NEST:** Excuse me, your Honor. This is far
16 outside what's --

17 **THE COURT:** Sustained. This is a speech. Sustained.

18 **BY MR. JACOBS:**

19 **Q.** Could you answer very briefly, Dr. Mitchell, what your
20 opinion is with respect to whether rangeCheck is found on
21 Samsung phones and why?

22 **A.** This code appears in the source code archive of Samsung
23 and, therefore, very likely appears on the Samsung phones.

24 **Q.** Thank you.

25 To sum up, Dr. Mitchell, what is your opinion with

1 respect to code-to-code copying between Java code and Android
2 code?

3 **A.** There's three kinds of examples that show direct code
4 copying. I think the most important factor there is that it
5 shows that this wasn't a clean room implementation where the
6 developers had no access and didn't look at the Java -- Oracle
7 Java code.

8 This shows examples that I don't -- I can't see any
9 other explanation than the developers had access and copied
10 portions of the code.

11 **Q.** In particular, with respect to decompilation,
12 Dr. Mitchell, could that happen by accident?

13 **A.** No, someone has to decide to do that. Basically, I have
14 to decide to cheat in a sense and produce your source code by
15 decompiling someone else's class file.

16 **MR. JACOBS:** Thank you very much, Dr. Mitchell.

17 **THE COURT:** All right. Cross-examination.

18 **MR. VAN NEST:** Give me just a moment to get set up,
19 your Honor.

20 (Brief pause.)

21 **CROSS EXAMINATION**

22 **BY MR. VAN NEST:**

23 **Q.** Good morning, Dr. Mitchell.

24 **A.** Good morning.

25 **Q.** The Java language is one of the most popular programming

1 language ever, right?

2 A. It certainly is very popular, yes.

3 Q. And it was released back in 1996?

4 A. I believed that's correct.

5 Q. And it became popular very quickly?

6 A. Yes.

7 Q. Thousands of developers were very soon writing the Java
8 programming language, correct?

9 A. Yes.

10 Q. They built a developer conference for the developers
11 trained in the language to come to every year called JavaOne?

12 A. Yes.

13 Q. And that became a larger and larger event every year?

14 A. Right. Yeah, I believe it was, even in those early years,
15 the largest developer conference at that time.

16 Q. And soon the Java language was being taught in colleges
17 and universities around the country --

18 A. Yes.

19 Q. (Continuing) -- right?

20 Thousands of students learning the Java language
21 every year?

22 A. Yes.

23 Q. You were teaching -- you were teaching the Java language
24 in your class?

25 A. I have covered it. It's also been covered in the -- in

1 other programming classes at our university and others.

2 Q. And so a large group of people writing in the Java
3 language has grown up over time and now there's a very large
4 group, hundreds of thousands of people writing in the Java
5 programming language around the world?

6 A. There are many developers who are familiar with the
7 language and use it.

8 Q. And the language itself has been among the top three
9 programming languages for a long time, right?

10 A. The statistics probably show that, yes.

11 Q. And you set forth in your report that even in 2011 the
12 Java programming language was number one in terms of popularity
13 of programming languages around the world?

14 A. You're probably referring to what's called of Tiobe
15 Survey. It's based on examination of web pages.

16 Q. And in preparing your report, you assumed that Oracle was
17 making no claim for copyright infringement based on the use of
18 the language, right?

19 A. I have heard that said, yes.

20 Q. Actually, you relied on it in your report, correct?

21 A. I wasn't asked to look into issues of copyright of the
22 language specifically.

23 Q. In other words, in preparing your opinions you started
24 from the premise that the Java language itself was free to use
25 and there was no claim in this case of infringement based on

1 using the language, right?

2 **A.** I don't think that's really a premise of anything I looked
3 at, but I understand that there has been some representation to
4 that effect.

5 **MR. VAN NEST:** Could we have Paragraph 147 of TX 686
6 up?

7 **BY MR. VAN NEST:**

8 **Q.** Do you have your report before you?

9 **A.** Yes, I do.

10 **Q.** Would you take a look at Paragraph 147, and it's TX 686.

11 (Witness complied.)

12 **MR. VAN NEST:** Your Honor, I understand the report is
13 not coming into evidence. May I display the report?

14 **THE COURT:** Just a second. Let me get there.
15 Yes, you may display that and read it into the
16 record.

17 (Document displayed)

18 **BY MR. VAN NEST:**

19 **Q.** Paragraph 147, that's a paragraph out of your opening
20 report Dr. Mitchell?

21 **A.** That's correct.

22 **Q.** And you said:

23 "I understand that Oracle is not asserting
24 copyright infringement of the Java language
25 as such, but of creative works that are

1 written in and using the Java language."

2 **THE COURT:** Java programming language.

3 **MR. VAN NEST:** Java programming language.

4 **BY MR. VAN NEST:**

5 **Q.** Is that the assumption that you made in preparing your
6 report?

7 **A.** That's what I was told, yes.

8 **Q.** And were you here in the courtroom when Mr. Ellison
9 testified that nobody owns the Java programming language?

10 **A.** I heard his testimony.

11 **Q.** And that the Java programming language can be used by
12 anyone without any royalty; that's what he said, right?

13 **A.** I heard what he said, yes.

14 **Q.** Now, the Java programming language specification was
15 actually published back in 1996, correct?

16 **A.** That sounds right.

17 **Q.** And that was a publication by Sun of the language and all
18 the specification of the language?

19 **A.** I believe there is a book that contains that information
20 if that's what you're referring to.

21 **Q.** And are you familiar with the first edition of the *Java*
22 *Programming Language Specification*?

23 **A.** Yes, I am.

24 **Q.** And that's TX 2564.

25 **MR. VAN NEST:** Can I have 2564?

1 Actually, actually I have it here. May I approach
2 the witness, your Honor?

3 (Whereupon, document was tendered
4 to the witness.)

5 **BY MR. VAN NEST:**

6 **Q.** This was in your box.

7 Is that the first edition of the *Java Programming*
8 *Language Specification*?

9 (Document displayed)

10 **THE COURT:** Is this in evidence? It's on the screen.

11 **MR. VAN NEST:** I believe it is, your Honor.

12 **THE COURT:** All right.

13 **THE CLERK:** I don't show it in evidence.

14 **BY MR. VAN NEST:**

15 **Q.** Let's back up a minute and just ask: Are you familiar
16 with the specification, Dr. Mitchell?

17 **A.** Yes, I am.

18 **Q.** Okay. And that's the first edition of the *Java Language*
19 *Specification* that Sun published back in the 90's?

20 **A.** I believe so. I normally have a bound copy, but it looks
21 like a reproduction of that.

22 **MR. VAN NEST:** I'd move 2564 into evidence, your
23 Honor.

24 **THE COURT:** Any objection?

25 (No response.)

1 **THE COURT:** Hearing none, received.

2 (Trial Exhibit 2564 received
3 in evidence)

4 **BY MR. VAN NEST:**

5 **Q.** Now, this is a several hundred page document, right?

6 **A.** The first page suggests 852 pages.

7 **Q.** And was this part of an effort by Sun to make the language
8 out there popular and available to people?

9 **A.** I don't know fully their motives. This is clearly a
10 description of the language that would be useful to someone who
11 wanted to use it.

12 **Q.** And this was clearly published by the folks at Sun?

13 **A.** It's a book. You can buy it, yes.

14 **Q.** Did you make any investigation as to why Sun would be
15 publishing the *Java Language Specification* back in 1996?

16 **A.** Not really, no.

17 **Q.** It was to make it available so people could use the
18 language, right?

19 **A.** It certainly would be useful for that. I'm sure the
20 authors felt they would get recognition they deserved from
21 publishing the book, and there may be other reasons.

22 **Q.** And this was said to be a complete specification of the
23 syntax and semantics of the Java language and the core packages
24 java.lang, java.io, Java.util of the API, correct?

25 Actually, I'm sorry.

1 **MR. VAN NEST:** Could we put Page 23 up?

2 (Document displayed)

3 **BY MR. VAN NEST:**

4 **Q.** I'm sorry, Dr. Mitchell. It's a large book. I have it
5 actually on the screen. You can look at it in the text, but I
6 have it to the right on the screen.

7 **A.** Okay. Great. I know there is something on the back
8 cover, too. There is a description.

9 **Q.** Did you say back cover?

10 **A.** I think so.

11 **Q.** Actually, I'm looking at the preface, the very first page
12 of the preface. It's Page 23. Do you have it there on the
13 screen?

14 **A.** Yeah, I'll just look on the screen.

15 **Q.** That's in the preface, not the back cover, right?

16 **A.** That's what it looks like, yes.

17 **Q.** And in the very first volume of the language specification
18 some of these APIs you're telling us about, they are actually
19 included in the language specification, right?

20 **MR. JACOBS:** Objection, your Honor.

21 **THE COURT:** What's the objection?

22 **MR. JACOBS:** One thing, it's argumentative. The
23 document for -- for another, the document speaks for itself and
24 says quite clearly there is a syntax and semantics of the Java
25 language and the core packages.

1 **MR. VAN NEST:** This is cross-examination, your Honor.

2 **THE COURT:** This is cross-examination and the whole
3 point of an expert is to argue with him on cross.

4 (Laughter.)

5 **THE COURT:** Nothing they have said is a fact. It's
6 all opinion, and counsel is entitled to argue with those
7 opinions, and this is one way to do it.

8 This expert is able to protect himself. He's a
9 Stanford -- right? Aren't you at Stanford?

10 **THE WITNESS:** Yes. Other days of the week --

11 **THE COURT:** I'm sure he's an MIT, Stanford. He can
12 protect himself. The objection is overruled.

13 **MR. VAN NEST:** Especially against me, your Honor.

14 (Laughter.)

15 **THE COURT:** Possibly.

16 (Laughter.)

17 **THE COURT:** Please ask that question again,
18 Mr. Van Nest.

19 **BY MR. VAN NEST:**

20 **Q.** So my question, Dr. Mitchell is: This book is intended to
21 be a complete specification of the Java language, correct?

22 **A.** That's what the first half of the sentence says, yes.

23 **Q.** And in the book is the complete specification for three of
24 the Java APIs we have been talking about: java.lang, java.io
25 and java.util, right?

1 A. I think the book also covers the APIs, as the second half
2 of the sentence says.

3 Q. And these core packages, actually each one of them gets a
4 several hundred page specification in the language book, right?

5 A. Well, they are documented in this book.

6 Q. Let's turn now -- can we show Chapter 20?

7 (Document displayed)

8 Q. Java.lang, that's one of the APIs that's accused in this
9 case, correct?

10 A. It is one of the 37 APIs.

11 Q. But it's fundamental to using the language, isn't it?

12 A. There are various ways to evaluate the degree to which
13 this is fundamental. And according to -- depending on which
14 criteria you apply, you generally find some portion of
15 java.lang.

16 Q. In other words, without any of java.lang, the API, the
17 language really doesn't work; isn't that; right, Dr. Mitchell?

18 A. There are a few things, such as Object, that are in here
19 that I think are essential to the language.

20 Q. So without java.lang, the language doesn't work because
21 java.lang has things in it that are essential to the language,
22 right?

23 A. Yes. There are also some things that might not be so
24 essential to the language.

25 Q. Now, this particular chapter is 100-some pages long,

1 specifies the entirety of the java.lang API, correct?

2 **A.** I believe that's correct.

3 **MR. VAN NEST:** And if we could see Chapter 21?

4 (Document displayed)

5 **BY MR. VAN NEST:**

6 **Q.** Chapter 21 that's java.util, that's another API package
7 that's accused in this case, correct?

8 **A.** Correct.

9 **Q.** That's part of this book on the java.lang specification,
10 right?

11 **A.** This chapter is there in the book and it is the "Java
12 Language Specification."

13 **Q.** And the Java language wouldn't be much use without
14 java.util, would it?

15 **A.** I think you could write some things in the language
16 without using many portions of java.util and perhaps without
17 using any of it.

18 **Q.** But without java.util there's a lot of things that someone
19 using the language would expect to be there that wouldn't be
20 there, right?

21 **A.** Well, certainly, the library and its packages are useful
22 in programming.

23 **Q.** Now, java.io is also specified in its entirety in this
24 book? That's Chapter 22.

25 (Document displayed)

1 Q. Correct?

2 A. My understanding is this is a description of java.io at
3 the time the book was published.

4 Q. Okay. And these three chapters are described in this
5 reference manual as:

6 "...the core of the Java API which must be
7 included in all Java general purpose
8 systems."

9 Right?

10 MR. VAN NEST: Can we go to Page 5 of the
11 introduction?

12 (Document displayed)

13 BY MR. VAN NEST:

14 Q. Do you have that, Dr. Mitchell?

15 A. Yeah. I think that's saying that in a Java installation,
16 these APIs and associated packages and library are part of this
17 standard or normal installation.

18 Q. So what the text actually says is:

19 "These packages" -- and that's referring to
20 java.lang, java.util, java.io -- "must be
21 included in all general purpose Java
22 systems."

23 Right?

24 A. In Java systems. That's literally what the sentence says,
25 yes.

1 Q. And if we go down a little bit to the next page -- excuse
2 me, not the next page. The next paragraph.

3 That describes java.lang, which you said contained
4 elements essential to the language. The book says:

5 "The types defined in java.lang are
6 automatically imported to be available
7 without qualification in all Java programs."

8 Right?

9 A. Yeah. This is describing the way that elements of
10 java.lang can be named in programs without an explicit
11 statement to import them.

12 Q. And then Chapter 21 is described there:

13 "Java.util defines a few basic utility
14 classes, such as hashtable class and a
15 pseudo-random number generator."

16 Correct?

17 A. Yes, that's what the sentence says.

18 Q. Now, without java.io could a programmer writing in the
19 Java language even print something, send something to a
20 printer?

21 A. Printing is an IO feature, so I believe that requires some
22 elements of java.io, but a program could do something else.

23 Q. And without java.io could a programmer writing in the Java
24 language even send text to the screen of his laptop or desktop?
25 Doesn't that require IO, too?

1 A. I believe it may.

2 Q. Now, in 1996 Sun also published a book entitled *Core*
3 *Packages of the Java API*, correct?

4 A. I expect so.

5 Q. And do you recall that Java.net, in addition to java.io
6 and java.util and java.lang, was described as a core package?

7 You don't need to look at anything. I'm just asking
8 whether you are aware that java.net from the very beginning has
9 been described by Sun as a core package?

10 A. I believe that's the terminology. I don't -- I don't
11 recall exactly the way it was stated.

12 Q. And Sun refers to those core packages as fundamental to
13 the Java language?

14 A. I think the sentence that we saw before said fundamental
15 to the Java system or some other term besides "language."

16 Q. Well, let's take a look.

17 THE COURT: Is this in evidence?

18 BY THE COURT:

19 Q. Let's take a look --

20 THE COURT: Is 980 in evidence?

21 MR. VAN NEST: 980 is in evidence, I believe, your
22 Honor.

23 THE COURT: All right. Then go ahead. Why don't we
24 put it on the screen so everyone can see it.

25 MR. VAN NEST: Can we have 980, the back cover --

1 let's stay with the front here.

2 (Document displayed)

3 **BY MR. VAN NEST:**

4 **Q.** This is also 1996, correct?

5 **A.** Yes.

6 **Q.** The *Java Application Programming Interface, Volume 1, Core*
7 *Packages*, correct?

8 **A.** That's what it says on the book, yep.

9 **MR. VAN NEST:** And if we could go to the back cover?

10 (Document displayed)

11 **MR. VAN NEST:** Let's go right there to the second --
12 there we go. No, let's go up a little bit, "Java Interface."

13 **BY MR. VAN NEST:**

14 **Q.** (As read)

15 "Volume 1 Core Packages describes the
16 libraries that are the foundation of the Java
17 language."

18 That's what it says, right?

19 **A.** Yes.

20 **Q.** "The foundation of the Java language." Were you familiar
21 with that, with that phrase?

22 **A.** I see it now. I'm sure I've read it at some previous
23 point.

24 There are various descriptions at this time about the
25 relationship between portions of the libraries and the

1 language.

2 **Q.** And it goes on to say:

3 "These are the general-purpose libraries
4 fundamental to every Java program."

5 Right?

6 **A.** Yes.

7 **Q.** And the core package -- if you turn to the table of
8 contents, it's about five pages in. You have it there before
9 you, Mr. Mitchell.

10 The core packages in Volume 1 include java.lang,
11 java.io, java.util and java.net, correct?

12 **MR. VAN NEST:** Let's go to the table of contents, if
13 we could, Ben?

14 **BY MR. VAN NEST:**

15 **Q.** I have it as VII right at the front.

16 (Document displayed)

17 **MR. VAN NEST:** There we go. Let's go to the first
18 page there.

19 **BY MR. VAN NEST:**

20 **Q.** Do you have it? Java.lang, that's one of the core
21 packages that's described as the foundation of the language,
22 right? That's in this volume.

23 **MR. JACOBS:** Your Honor, objection. Scope.

24 **THE COURT:** What do you mean scope? Why isn't this
25 within the scope of the direct?

1 **MR. JACOBS:** Because we never discussed the
2 relationship between the packages and the language in his
3 direct testimony.

4 **THE COURT:** Well, I understand possibly that, but
5 this is cross-examination and I think this line of questions
6 generally relates to the points that were made on direct
7 examination. So even though it's not exactly within the scope
8 of the direct, it's close enough.

9 So the objection is overruled. Please continue.

10 **BY MR. VAN NEST:**

11 **Q.** So java.lang, that's one of the core packages as defined
12 by Sun itself, right?

13 **A.** This is one of the packages that in -- in one of the
14 passages you read I think is referred to as a core package.

15 **Q.** Well, actually, all the packages in this volume are
16 described as the foundation of the Java language. We saw that
17 on the back cover, right?

18 **A.** Yes, that's what the sentence on the back cover says.

19 **MR. VAN NEST:** Okay. Then let's go to the second
20 page, java.io.

21 (Document displayed)

22 **BY MR. VAN NEST:**

23 **Q.** That's in this volume, right?

24 **A.** Yes.

25 **Q.** Part of the foundation of the language, correct?

1 A. In the same way, yes.

2 Q. And java.util?

3 THE COURT: I'm confused. Is the witness
4 disagreeing?

5 Do you disagree with those statements in the book,
6 that it's fundamental and so forth?

7 THE WITNESS: I believe there are various ways to
8 understand the relation between the API and the language. I
9 think there are also from this time frame different statements
10 and different references.

11 THE COURT: Well, as of the time these books were
12 published, would you have agreed with those statements or are
13 you saying that they are not true?

14 You keep saying that that's what the book says, like
15 you don't agree with it.

16 THE WITNESS: I think there are other ways to
17 understand the relationship --

18 THE COURT: Can't you say "yes" or "no"? Do you
19 disagree with what's written there as of the time it was
20 written?

21 THE WITNESS: I certainly don't disagree that that's
22 what's written.

23 THE COURT: Do you disagree with the truth of it?

24 THE WITNESS: I believe there are other statements
25 that add some nuance and cast different light on this topic.

1 **THE COURT:** All right. So you disagree with it?

2 **THE WITNESS:** In that way, yes. To that extent.

3 **THE COURT:** All right. Go ahead, counsel.

4 **BY MR. VAN NEST:**

5 **Q.** Now, java.net is needed in order to -- for someone writing
6 in the Java programming language to get access outside his or
7 her computer to a network, right?

8 **A.** This is one package that provides networking. And at this
9 time, this version that may have been the only package that did
10 so.

11 **Q.** Okay. And so without it, a programmer writing in the Java
12 programming language couldn't even get out and network to
13 another computer, right?

14 **A.** I believe that's correct.

15 **Q.** It's not correct or it is correct?

16 **A.** It's correct, as far as the information --

17 **Q.** Okay, it is correct.

18 And the whole point of the Java language was to allow
19 networking type applications, right?

20 **A.** I don't think that's correct. The Java applets that were
21 one of the popular uses of Java did not need to network back to
22 the site they came from. They were downloaded over the network
23 through the actions of the web browser, but not necessarily
24 through actions implemented in the -- or programmed in the Java
25 language.

1 Q. But at the time this book was written, it was necessary to
2 use java.net to do any networking at all, right?

3 A. As far as I know, yes.

4 Q. Okay. Now, there is a third edition of the language that
5 came out a few years later, correct?

6 A. Yes.

7 Q. And that's Exhibit 984.

8 MR. VAN NEST: It's already in evidence, your Honor.
9 We saw it last week.

10 (Document displayed)

11 BY MR. VAN NEST:

12 Q. This is the *Java Language Specification, Third Edition*.
13 Are you familiar with this?

14 A. Yes, I am.

15 Q. And I think, I have the book right here.

16 MR. VAN NEST: May I approach the witness, your
17 Honor?

18 (Whereupon, book was tendered
19 to the witness.)

20 BY MR. VAN NEST:

21 Q. Now, the third edition talks about certain classes that
22 have a special relationship to the language, is that right?

23 A. I'm sure there is some phrase like that somewhere.

24 Q. Let's take a look at Page 6.

25 MR. VAN NEST: Display that on the screen.

1 (Document displayed)

2 **MR. VAN NEST:** There we go.

3 **BY MR. VAN NEST:**

4 **Q.** All right. This describes in the second sentence that:

5 "Some classes have a special relationship

6 with the Java programming language."

7 Do you see that?

8 **A.** Yes.

9 **Q.** And it actually lists a bunch of classes and interfaces in
10 java.lang.reflect, among others?

11 **A.** Yes.

12 **Q.** Now, java.lang.reflect, that's another package beyond
13 java.lang, java.io, java.net, correct?

14 **A.** Yes.

15 **Q.** That's another API package. That's also accused in this
16 case, right?

17 **A.** Yes.

18 **Q.** Yes, okay. And the Objects classes, ClassLoader, String,
19 Thread, those are all necessary in order for the Java language
20 to operate properly, correct?

21 **A.** I believe those are -- Thread is essential for current
22 programs, not essential for sequential code.

23 **Q.** But Class, that Class is essential to operate the
24 language, correct?

25 **A.** Yes. Object, Class, ClassLoader, yes.

1 Q. All of those. And those are in this additional API,
2 java.lang.reflect, right?

3 A. I don't think Object, Class and ClassLoader are in the
4 Reflect package.

5 Q. Okay, fair enough. Some are in java.lang, but there are
6 additional classes and interfaces in package java.lang.reflect
7 that have this special relationship, right?

8 A. I think the ones in reflect are for what's called
9 reflection. That's not essential for most Java programming.

10 Q. Now, last week were you here when Dr. Reinhold identified
11 61 classes that were mentioned in the specification as part of
12 the specification?

13 A. I'm aware of that list, yes.

14 Q. Did you make any effort in forming your opinion to
15 determine how many classes were actually necessary to operate
16 the method -- excuse me, operate the language, as Dr. Reinhold
17 did?

18 MR. JACOBS: Your Honor, this is now far beyond the
19 scope.

20 THE COURT: Overruled.

21 A. I discussed that with Dr. Reinhold, and I understand and
22 agree with his analysis.

23 BY MR. VAN NEST:

24 Q. So you understand and agree with Dr. Reinhold that there
25 are at least 61 classes needed to operate the Java language

1 itself, right?

2 **A.** I think the criterion he used was based on the
3 specification, taking the point of view that the specification
4 defines the language.

5 **Q.** Okay.

6 **MR. VAN NEST:** Could we have TX 1062, please?

7 In this is in evidence, your Honor.

8 (Document displayed)

9 **MR. VAN NEST:** And just highlight the top portion of
10 the page, please.

11 **BY MR. VAN NEST:**

12 **Q.** Do you have 1062 up there? I probably have it in the box
13 here.

14 Can you see it on the screen? It's just a one-page
15 exhibit.

16 **THE COURT:** We now have it blown up, so we need to
17 shrink it back down to the whole page.

18 **MR. VAN NEST:** Okay. Shrink it back down.

19 **THE COURT:** Can you see it?

20 **THE WITNESS:** Yes. Thank you.

21 **THE COURT:** Let's leave it on the whole page since
22 you don't have the trial exhibit.

23 **MR. VAN NEST:** Let's find the trial exhibit. I think
24 we do have it, your Honor.

25 (Brief pause.)

1 **MR. VAN NEST:** We'll pull it.

2 **THE COURT:** Let's finish up this exhibit and then
3 we're going to take a break.

4 **BY MR. VAN NEST:**

5 **Q.** In any event, you agree with Dr. Reinhold's analysis last
6 week that these 61 classes are all part of the Java language
7 specification, correct?

8 **A.** I understand the basis on which he did this analysis. And
9 if you take the language specification to be the definition of
10 the language, then the conclusion from that starting point is
11 this list of 61 classes.

12 **MR. VAN NEST:** Your Honor, this is a good time to
13 break.

14 **THE COURT:** All right. Please remember the
15 admonition.

16 Ms. Michals, could you stay behind for just a moment?
17 Which one is Ms. Michals?

18 (Ms. Michals raises her hand.)

19 **THE COURT:** Is that who you meant?

20 **THE CLERK:** No, that's not. I had them switched
21 around.

22 **THE COURT:** Your note says Ms. Michals. Did you mean
23 Ms. Gallo?

24 **THE CLERK:** It was Ms. Gallo.

25 **THE COURT:** Ms. Gallo, raise your hand.

1 (Ms. Gallo raises her hand.)

2 **THE COURT:** Is that who you meant?

3 **THE CLERK:** Yes.

4 **THE COURT:** Ms. Gallo, I apologize. The seating --
5 we need to ask you to stay behind for just a moment. It will
6 just take 60 seconds.

7 Everyone else just remember the admonition. Do not
8 speculate about why we want Ms. Gallo to stay behind.

9 (Laughter.)

10 **THE COURT:** Okay. You can go into the jury room.

11 (Jury exits courtroom at 11:13 a.m.)

12 **THE COURT:** The witness can step down and go out and
13 take his break, too.

14 **THE WITNESS:** Okay.

15 **THE COURT:** Thank you sir. Go ahead.

16 (Witness steps down.)

17 **THE COURT:** Okay. Everyone else be seated.

18 We're going to need the microphone maybe.

19 Ms. Gallo, Dawn told me that you had raised with her
20 an issue about your employer, okay? Now, I need -- before we
21 go any further with that, I need to explain to you how I'm
22 required to deal with that, if there is an issue.

23 The first thing I would need to do is inquire further
24 here in open court about what the problem is and to then --
25 after you're out of our presence, I will consult with the

1 lawyers about how to deal with it, if we can.

2 So I just wanted you to be aware that I'm willing to
3 hear out what the issue is, but I can't promise you that I can
4 solve it, and I'm not sure I can say to you whether what your
5 employer wants to do is legal or not legal. Sometimes there
6 may be an easy answer, but I would have to know a lot more than
7 what Dawn wrote down on this little piece of paper.

8 So I don't want to go further with this unless you
9 want me to. That's why -- and I'm happy to try. It's just I
10 need to tell you that it's not as simple as a private
11 conversation between you and me or you and Dawn. You're now an
12 official. You're an official member of this jury and I have to
13 do this in open court in the presence of the public and the
14 lawyers.

15 So you want to take some time to think about whether
16 you want to do that? If you do, that's fine. We'll have a
17 little inquiry. See what I mean?

18 **JUROR GALLO:** I mean, it's very complicated, so I
19 don't want to waste anybody's time. I just had a couple of
20 questions. I'm still trying to work it out with my employer,
21 so if --

22 **THE COURT:** If you -- it's up to you.

23 **JUROR GALLO:** Why don't I just talk to them and see
24 if we can work it out. Then if for some reason we can't come
25 to an understanding, I'll let you know then.

1 **THE COURT:** I will say this much. I don't know the
2 problem. You know, when the hardship questionnaire went out, I
3 specifically said you must check with your employer as to
4 whether or not -- how long they will pay for jury service,

5 **JUROR GALLO:** Right.

6 **THE COURT:** And everyone on the jury was supposed to
7 check that and know the answer.

8 So now if your employer told you one thing back then
9 and now has changed their mind, then I may be able to do
10 something about that.

11 **JUROR GALLO:** I did ask them. It relates to
12 benefits, which they didn't mention, so... They are saying a
13 different -- they didn't say anything at all before and now
14 they are saying something, so I'm just trying to figure it out.

15 **THE COURT:** Well, since we've gone this far, do you
16 want to just tell me what the issue is? It's up to you. If
17 you want to drop it and go talk to your employer, that's okay,
18 too, and you can bring it back up later, but I -- I just can't
19 talk with you in the hallway about it.

20 **MR. VAN NEST:** Your Honor, we would be happy to do it
21 at sidebar, if the Court felt there wasn't a reason to do it in
22 open court.

23 **THE COURT:** Would you like to do that at the sidebar?

24 **JUROR GALLO:** Okay.

25 **MR. JACOBS:** We agree, your Honor.

1 **THE COURT:** What?

2 **MR. JACOBS:** We agree. Sort of private.

3 **THE COURT:** All right. Well, why don't you come over
4 to the sidebar?

5 (UNDER SEAL proceedings
6 were held at sidebar.)

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

(UNDER SEAL proceedings
were held at sidebar.)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

(UNDER SEAL proceedings
were held at sidebar.)

1 (UNDER SEAL proceedings
2 were held at sidebar.)
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

22 (Whereupon, the following proceedings were
23 held in open court, in the presence and
24 hearing of the jury.)

25 **THE COURT:** All right. Well, just one question. Do

1 you all have a truce among you not to inquire about the amount
2 of money experts have been paid?

3 **MR. VAN NEST:** There is no truce, your Honor, that
4 I'm aware of.

5 **THE COURT:** Well, it cuts both ways. So sometimes
6 the lawyers do that, but all right.

7 Anything the lawyers need the Court for?

8 **MR. VAN NEST:** No, your Honor.

9 **MR. JACOBS:** No, your Honor.

10 **THE COURT:** We'll take a 15 minutes ourselves. Thank
11 you.

12 (Whereupon there was a recess in the proceedings
13 from 11:22 a.m. until 11:37 A.M.)

14 **THE COURT:** All right. Be seated. Let's go back to
15 work.

16 Before I bring in the jury, I just want you to be
17 aware that -- as I know you know this, but when you -- on
18 redirect examination the witness is not limited to the expert
19 report. He is free to say anything.

20 You have very broad-ranging questions. He's free to
21 go to town on redirect on all of your questions, even though --
22 because you're way beyond what's in the report. So he's not
23 limited on redirect to the report. You --

24 **MR. VAN NEST:** I understand, your Honor.

25 **THE COURT:** You understand that, okay. All right.

1 **MR. VAN NEST:** I would assume if there is an area
2 that I've objected to and you said, yeah, it's outside the
3 report and I don't go into it on cross, he can't get up and do
4 it on redirect either.

5 **THE COURT:** Well, that part is correct, but
6 everything that you've covered so far, Mr. Jacobs can come back
7 and say, "All right, with respect to that subject..." You
8 know, to make the counter points.

9 And even whether it's in the report or not, that's
10 just the way it has to work.

11 **MR. VAN NEST:** Understood.

12 **THE COURT:** It wouldn't be fair otherwise.

13 **MR. VAN NEST:** Understood.

14 **THE COURT:** All right. I'm going to bring the jury
15 in.

16 (Jury enters the courtroom at 11:38 a.m.)

17 **THE COURT:** Thank you. Be seated.

18 Everybody ready over there?

19 (Jury nodding affirmatively.)

20 **THE COURT:** Great. Please continue, Mr. Van Nest.

21 **MR. VAN NEST:** Thank you, your Honor.

22 **BY MR. VAN NEST:**

23 **Q.** Dr. Mitchell, you have described APIs as a set of names
24 that can be used to access features of the library together
25 with specified conventions about their use, correct?

1 A. Yes. That's correct. The specified conventions can be
2 extensive.

3 Q. And this set of names that can be used to access features
4 of the library, you're referring to the Package, Class, Method
5 names and that sort of thing?

6 A. All of the things that a programmer might include in their
7 program in order to access code written in the library.

8 Q. And you need to use these API elements, a programmer does,
9 to access source code, for example, in the libraries, correct?

10 A. No.

11 Q. The programmers looking to access prewritten code, I think
12 you said in your direct exam, correct?

13 A. Yes.

14 Q. All right. And that prewritten code, source code, is
15 compiled and then the compiled code is in the library --

16 A. Yes.

17 Q. (Continuing) -- right?

18 And so one of the functions of these API names is to
19 use them so that the programmer can get access to the source
20 code and eventually to the compiled code in the library, right?

21 A. Not exactly. The source code is not accessed through
22 these names.

23 Q. But the names, you said, can be used to access features of
24 the library together with specified conventions, right?

25 A. That's what I said, yes.

1 Q. Let's actually take a look at -- you have your report
2 before you, TX 686, Paragraph 52.

3 MR. VAN NEST: If we could display that on the
4 screen, please?

5 THE COURT: For the record, since the report is not
6 in evidence and won't be, you can read the relevant part into
7 the record.

8 (Document displayed)

9 MR. VAN NEST: Let's highlight there, "An API
10 consists of a set of names." Let's go just to the right there,
11 Ben? Right in the middle of the paragraph it starts on the
12 right.

13 BY MR. VAN NEST:

14 Q. (As read)

15 "An API consists of a set of names that can
16 be used to access features of the library
17 together with the specified conventions about
18 their use."

19 That's one of the definitions you gave of an API?

20 A. That's a description. It's in my report.

21 Q. Okay. And the API names you're talking about there are,
22 for example, the Package name, the Class name and the Method
23 name, possibly among others, right?

24 A. Yes. Those are entities in the library.

25 Q. And a programmer writing in the Java language is going to

1 rely at some point upon prewritten code, right?

2 **A.** That's what using the library is, yes.

3 **Q.** That's what using the library is.

4 And down here in Exhibit 3452, which we saw last week
5 with Dr. Bloch, we're seeing source code here at the bottom.

6 He calls it implementation. Do you see that?

7 **A.** Yes.

8 **Q.** That's the -- that's the prewritten code that the
9 developer would be seeking access to?

10 **A.** Again, the developer may wish to access the compiled code
11 produced from that.

12 **Q.** He would generally do it through this source code,
13 correct?

14 **A.** No.

15 **Q.** Or he would access the compiled code, but this source code
16 is -- eventually becomes the compiled code?

17 **A.** In some order.

18 **Q.** Okay. Well, when you said features of the library, what
19 were you referring to?

20 **A.** Classes, Methods, Interface defined in the library.

21 **Q.** Okay. Now, a programmer writing in the Java language must
22 use the names `java.lang.Math.max` in order to invoke the
23 prewritten code that performs that max function, correct?

24 **A.** Yes.

25 **Q.** And, therefore, these names actually have a functional

1 purpose, right?

2 **A.** The names are used exactly as I described; to access the
3 compiled output from the prewritten code.

4 **Q.** And that's a functional purpose, right, correct?

5 **A.** Well, when these names are used in a developer's code, it
6 invokes the code from the library.

7 **Q.** Right. So when these names are used in the code, it
8 invokes the code in the library --

9 **A.** Yes.

10 **Q.** (Continuing) -- correct?

11 And in order to invoke the code in the library, the
12 developer has to use the names and he has to use them
13 correctly?

14 **A.** Yes. That's how the Java language and other programming
15 languages work.

16 **Q.** So, for example, if a developer wants to access the code
17 that performs the Math method -- the max method. He or she
18 must write java.lang.Math.max in his or her program?

19 **A.** In some form. Something equivalent, I think, to what you
20 said, but not literally that --

21 **Q.** If, for example, they used java.lang.Math.maximum, the
22 code wouldn't work unless that API had been rewritten, correct?

23 **A.** You cannot call the max function by writing maximum
24 instead.

25 **Q.** Cannot?

1 A. Correct.

2 Q. Right. So, and as a matter of fact, these conventions are
3 so strict that the max has to be lower case, correct? Small m?

4 A. The cases must match, yes.

5 Q. So max would typically be lower case, small m?

6 A. Typically.

7 Q. And Math, the class, would typically be upper case,
8 capital M?

9 A. There's a naming convention that is as you describe.

10 Q. And if I'm a programmer and I'm writing in the language
11 and I mess that up and I use math, lower case, and Max, upper
12 case, again, my code will not work?

13 A. For better or for worse, computers are very literal.

14 Q. And that would be true of every single method in every
15 single class throughout these Java APIs you have been
16 discussing, correct?

17 A. Yes.

18 Q. Without using exactly the same name, the code will not
19 work.

20 And if a platform provides a set of APIs that are
21 different from these -- either lower case, upper case,
22 different words, different names -- existing code that
23 programmers have written in the free Java language won't work
24 on that platform, right?

25 A. Code written for one API will not generally work for

1 another unless the names are the same and other features are
2 the same.

3 **Q.** So if, for example, I were to take max and take it out of
4 the Math class and put it in a different class, existing code
5 written by a programmer would not run properly because the name
6 would be wrong?

7 **A.** Yes. Some syntactic adjustment would be needed.

8 **Q.** Now, you've actually described these elements, the names
9 and the elements, package names, class names, interface names,
10 method names, as parts of speech, right?

11 **A.** I believe I gave some description in which the categories
12 such as the category of class names, is analogous to, say, the
13 category of nouns in natural language.

14 **Q.** You actually went further than that. You described the
15 elements of an API -- the package names, class and interface
16 names, relationships and methods -- as like parts of speech,
17 right?

18 **A.** There's a way to understand that accurately, and there's a
19 way to misunderstand it. So I was trying to clarify an answer
20 to your previous question.

21 **MR. VAN NEST:** Your Honor, I would like to play at
22 this time from Dr. Mitchell's deposition Page 120, Line 18
23 through 121, Line 10 -- excuse me, maybe start at Line 14 or
24 line 18?

25 **BEN BOTELLO:** 18.

1 **MR. VAN NEST:** I was correct.

2 **THE COURT:** All right. Go ahead.

3 (Videotape played in open court; not reported.)

4 **BY MR. VAN NEST:**

5 **Q.** Do you stand by that testimony, Dr. Mitchell?

6 **A.** I definitely said that. I think I explained further in my
7 report what that analogy.

8 **Q.** And using the parts of speech, java.lang.Math.max, you
9 must be precise in the Java language. If the parts of speech
10 change, the code you've written will not work; isn't that
11 right?

12 **A.** I don't think you're changing the parts of speech. You
13 are changing specific nouns.

14 **Q.** If you change specific nouns, the code will not run,
15 right? If I call max maximum, prewritten code in the Java free
16 language will not run, right?

17 **A.** As simple as if I forget your name and call you something
18 else, you may not respond.

19 **Q.** And in a computer, that means the program doesn't work?

20 **A.** That's correct.

21 **Q.** Now, I want to talk about your chart here for a minute. I
22 guess this is a representation of these Java API packages, is
23 that right?

24 (Document displayed)

25 **A.** It's an illustration that highlights certain features of

1 the packages and classes which are contained.

2 **Q.** And this illustrates the system in which they are
3 organized, right? The packages, the classes, the methods and
4 so on?

5 **A.** Yeah. It illustrates some of the hierarchal and
6 non-hierarchal relationships.

7 **Q.** So, for example, I have java.net right up here in the
8 middle of the top. Maybe I'll --

9 **MR. VAN NEST:** Can I move it a little bit closer,
10 your Honor?

11 **BY MR. VAN NEST:**

12 **Q.** Java.net, that's a package name, correct?

13 **A.** Correct.

14 **Q.** That's one of the 37 API packages that are accused in this
15 case, right?

16 **A.** Yes.

17 **Q.** And then organized within that package are the classes,
18 right --

19 **A.** Yes.

20 **Q.** (Continuing) -- is that correct?

21 **A.** Yes.

22 **Q.** So I think you said in your testimony the classes are
23 organized into packages. That means these classes are
24 organized under java.net?

25 **A.** Each class can be a member of at most one package.

1 Q. And typically the classes are listed not only under
2 java.net, but everywhere else here in alphabetical order,
3 right?

4 A. They may be listed that way on the poster.

5 Q. And they are listed that way generally in the
6 specification as well?

7 A. I don't recall, but that may be.

8 Q. And then there would be organized within each -- within
9 each class there would be methods?

10 A. Yes. The class has a set of methods.

11 Q. And those methods, typically in the specification, are
12 listed alphabetically within each class, right?

13 A. I don't think that's the case, no.

14 Q. Now, the methods are organized into classes and the
15 classes are organized into packages. That's the organization
16 of the Java language, right?

17 A. That's part of the organization.

18 Q. And this organization of packages, classes and methods in
19 Android was published by Google back in 2007, correct?

20 A. I don't know.

21 Q. The Android project appeared on a website in the fall of
22 2007, Dr. Mitchell, isn't that correct?

23 A. I don't have any recall of that. That may be.

24 Q. Certainly, the organization of the APIs is not something
25 you discovered once you were retained as an expert, correct?

1 **A.** The organization is evident from the poster and in other
2 ways.

3 **Q.** And in Android, the organization is evident and has been
4 for many years?

5 **THE COURT:** Let's make sure the witness -- see,
6 you're holding up, isn't that a Java chart?

7 **MR. VAN NEST:** It is, your Honor. Fair point.

8 **THE COURT:** But you're talking about Android. Let's
9 make sure the witness is on the same page.

10 I think counsel switched to Android, right?

11 **MR. VAN NEST:** I did.

12 **THE COURT:** So let's ask that question again to be
13 sure the witness had in mind Android as opposed to Java.

14 **BY MR. VAN NEST:**

15 **Q.** The organization and structure of the 37 APIs in Android
16 was published on a website back in 2007?

17 **A.** I didn't check the date. I don't recall, but I believe
18 that's completely possible.

19 **Q.** And, certainly, the organization structure in Android is
20 something that anyone that wanted to could determine back as
21 far as 2007, right?

22 **A.** Well, whenever this was available on the web. And someone
23 else could have looked at it just the same way I did.

24 **Q.** Wasn't hard to find. It wasn't hard to see. It wasn't
25 hard to determine what was there, correct?

1 A. The web is easily available from any network computer.

2 Q. Now, as part of your assignment, you testified that you
3 did some code-to-code analysis to determine whether code in
4 Android had been copied from Java?

5 A. Yes.

6 Q. And you actually relied on another expert to help with you
7 that, is that right?

8 A. Another expert identified these examples, but once they
9 were identified, I looked at them myself.

10 Q. And that expert is Dr. Visnick, Dr. Visnick?

11 A. There are two experts. I believe Dr. Visnick identified
12 some of the examples.

13 Q. And you relied on Dr. Visnick's report?

14 A. I read the report. I used some information from it. But
15 as I said, when I saw the examples, I looked at them myself.

16 Q. Fair enough. But he did an exhaustive analysis of all of
17 the Android source code in the Android libraries, right?

18 A. I believe he was asked to do that, yes.

19 Q. And he analyzed hundreds of thousands of lines of code?

20 A. Yes.

21 Q. You reviewed a report that he prepared?

22 A. Yes.

23 Q. Is that Exhibit -- I believe it's Exhibit 727. Is that
24 one of the things you relied on in forming your opinions in the
25 case, Dr. Mitchell?

1 A. If the number is correct, then, yes, I read his report and
2 I took the information from that.

3 MR. VAN NEST: May I approach?

4 THE COURT: Yes.

5 (Whereupon, document was tendered
6 to the witness.)

7 BY MR. VAN NEST:

8 Q. Do you recognize Exhibit 727?

9 A. Yes, I do.

10 Q. And is that the report of Dr. Visnick?

11 A. This is a report signed by Mark Visnick, yes.

12 Q. And you relied on it in forming your opinions as to
13 code-to-code copying, right?

14 A. I used this -- I read this report and I used the examples
15 to conduct my own study.

16 Q. And he did what he called a comparative textual analysis
17 comparing Android 2.2 to Sun's Java JDK, correct?

18 A. If that's what it says then, yes.

19 Q. Let's take a look.

20 MR. VAN NEST: Can we have Page 2 --

21 I would offer 727 into evidence, your Honor.

22 MR. JACOBS: Interesting. It's an expert report.

23 MR. VAN NEST: Actually, I will withdraw that
24 question. I understand these are probably not coming into
25 evidence.

1 May I display Page 2 of 727?

2 **MR. JACOBS:** No objection, your Honor.

3 **THE COURT:** All right. You can do that.

4 (Document displayed)

5 **BY MR. VAN NEST:**

6 **Q.** Page 2 gives an overview.

7 **MR. VAN NEST:** Let's go to the next page. I guess
8 it's the third page. There we go. Overview.

9 **BY MR. VAN NEST:**

10 **Q.** "Counsel for Oracle asked JLI" -- that's Dr. Visnick's
11 company?

12 **A.** I believe he works for that company.

13 **Q.** "...to use the source code repositories corresponding to
14 the following products to compare."

15 He compared Java JDK, that's the Sun version; right?

16 **A.** Yes.

17 **Q.** To Android 2.2; correct?

18 **A.** Yes.

19 **Q.** That ended up being a comparison of hundreds of thousands
20 of lines of code, right?

21 **A.** Yes.

22 **Q.** Maybe even millions?

23 **A.** Could be.

24 **Q.** And thousands and thousands and thousands of separate
25 files, right?

1 A. Yes. These are both very extensive.

2 Q. And he first applied what he calls a literal match
3 analysis of the two bodies of source code, correct?

4 Let's go down to Paragraph 16. It's at Page 5.

5 (Document displayed)

6 A. That's what it says, yes.

7 Q. And that literal match analysis --

8 MR. VAN NEST: Could we highlight the first two
9 paragraphs?

10 (Document highlighted)

11 BY MR. VAN NEST:

12 Q. That literal match analysis of the two bodies of source
13 code, that uses an automated tool, a special computer tool to
14 compare all of these thousands and thousands of lines of code.

15 A. That's what his report says, yes.

16 Q. And this particular analysis, he looked to see whether
17 there were identical lines of text in the two bodies of code,
18 right?

19 A. Yes.

20 Q. And then he did a second analysis in which he looked for
21 parts of a line of code that matched, is that right?

22 A. If that's what his report says, then I believe that's what
23 he did.

24 Q. Then once he did that he had a third analysis, what he
25 calls a token analysis, on the two bodies of source code to

1 determine even further whether there was any copying of words
2 or phrases in the code, right?

3 A. If you're referring to what he says is his third step,
4 that he says is a substring match analysis.

5 Q. Well, actually, let's get it up on the screen. Let's go
6 back to 17. That's actually the second step. He calls it
7 third, but ...

8 The substring match analysis uses another automated
9 tool to look for matching sequences or characters within lines
10 of text, right?

11 A. That's what his report says, yes.

12 Q. Then in paragraph 18 he describes a third analysis he did,
13 a token analysis, on the two bodies of source code?

14 A. Yes, that's what the report says.

15 Q. Is there any doubt in your mind that this was a very
16 thorough report and a very thorough examination?

17 A. He described exactly what he did here.

18 Q. And he looked -- he looked through all the lines of source
19 code in Android and all the lines of source code in the JDJ
20 1.5, correct?

21 A. I believe that's what he was asked to do. I have no
22 reason to think he didn't do it.

23 Q. And that ended up being thousands and thousands of files,
24 right?

25 A. Yes.

1 Q. And he came up, after that analysis and all the various
2 steps, and looking at all that code, he came up with 12 files
3 that you testified to us about this morning, correct?

4 A. He did identify those 12 files.

5 Q. And that's all he identified based on his thorough
6 analysis, right?

7 A. I believe those were the files that met the criteria he
8 was using in his search.

9 Q. And they, sir, were the only files that met the criteria
10 in his search, right?

11 A. I believe so.

12 Q. Okay. So we've got 12 files.

13 THE COURT: If you're going to draw on there again,
14 please use a magic marker. It's easier to see. Because that
15 color you selected is hard to see, I would use a darker marker.

16 MR. VAN NEST: Permanent marker.

17 THE COURT: That's good. Thank you.

18 MR. VAN NEST: Thank you, Your Honor.

19 (The court reporter interrupts. Pause in proceedings
20 in order for the reporter to repair her computer
21 malfunction.)

22 THE COURT: All right. We will now resume. And the
23 court reporter is back online.

24 Continue on, please.
25

1 BY MR. VAN NEST:

2 Q. So after Dr. Visnick did his analysis, he came up with 12
3 files out of the thousands and thousands of files that make up
4 Android, right?

5 A. These are the 12 files, I believe, or the 12 categories
6 that he found according to his search criteria.

7 Q. So two of those files were called "Timsort," right?

8 A. Slightly longer names on one of them, but, basically, you
9 could categorize them that way.

10 Q. I think you said one was Timsort and one was called Comp?

11 A. ComparableTimSort.

12 Q. TimSort. Okay. But there were two of those that you
13 testified about as being copies, correct?

14 A. Yes. There's a portion in those files that's copied.

15 Q. Now, the Timsort files, that represents nine lines of
16 code, right? Nine lines of code?

17 A. I think you're speaking of the rangeCheck function within
18 that.

19 Q. Good point. Timsort is the file in which rangeCheck was
20 found?

21 A. Yes.

22 Q. Fair enough. Okay.

23 A. Got it.

24 Q. So Timsort, itself, is a file of a thousand, 1500 lines of
25 code, something like that?

1 A. I don't remember, but it's bigger than nine.

2 Q. It's bigger.

3 And you looked -- Dr. Visnick looked at that file
4 among the many thousands of other files, and he didn't find
5 anything even in that file that looked like a copy, except for
6 the nine lines in rangeCheck, right?

7 A. I believe that's correct.

8 Q. And you have said that's a small amount of code?

9 A. Nine lines.

10 Q. And nine lines out of hundreds of thousands of lines,
11 right?

12 A. If you compare that to the larger library, then it's
13 exactly that ratio. Still nine lines.

14 Q. And were you here when -- strike that.

15 Dr. Bloch testified that he wrote the Java time in
16 which rangeCheck was originally filed, correct?

17 A. I believe that's correct, yes.

18 Q. And he wrote rangeCheck, which is the file now in Android
19 or was in Android, right?

20 A. I believe he's the author of that.

21 Q. So, in effect, he copied his own work. Is that what you
22 took from his testimony?

23 A. I understand that while he was working for Google he
24 copied the work that he had done previously working for Sun and
25 Oracle.

1 Q. Now, you know that now, today, rangeCheck is out, publicly
2 available. Sun has made rangeCheck free and open for anybody
3 to use, right? Are you aware of that?

4 A. No.

5 Q. No one has told you that rangeCheck is part of an open
6 source version of Java that anybody can use?

7 A. I believe there are licenses associated with the code
8 that's available on the Web.

9 Q. You don't know one way or the other, Dr. Mitchell, right?

10 A. No, but I don't understand that characterization at all.
11 What do you mean?

12 Q. Now, would rangeCheck itself, these nine lines of code in
13 a file called "Timsort," among millions of other lines of code,
14 would it have any independent economic value?

15 A. Independent of the library that it's in?

16 Q. Yes.

17 A. I think its main use is in the library. So I'm not sure
18 what the economic value of it is outside library. Certainly,
19 within the library it's useful.

20 Q. Now, Dr. Bloch, said rangeCheck could be written by a high
21 school programmer. Does that sound right to you?

22 A. Not really.

23 Q. How about one of your graduate students?

24 A. Uhm, a good high school programmer or graduate student, if
25 told exactly what was needed, could write the code.

1 Q. It's not that difficult, right?

2 A. Well, the code has some subtlety. And I think the
3 interesting part is figuring out exactly what you wanted the
4 function to do, more than realizing that function in Java code
5 once that's understood.

6 Q. Now, you mentioned two files that you referred to as
7 source code comments, Dr. Mitchell, correct?

8 A. Those are files from which the comments are copied.

9 Q. Okay. Comments in source code are never compiled when the
10 source code is compiled into bytecode, correct?

11 A. Approximately.

12 Q. If the source code comments were removed, it wouldn't make
13 any difference to the operation of the ultimate program, right?

14 A. Compiled bytecode does not contain the comments.

15 Q. So if you took the comments out, the compiled bytecode
16 would be the same?

17 A. Yes. Source code would be different. Someone reading the
18 code would see something different. Other programmers would
19 see something different. But someone using the compiled code
20 would have the same effect.

21 Q. Because these are English-language comments that are just
22 intended to give some guidance to programmers reading source
23 code, right?

24 A. Yeah, comments are for other programmers or users of the
25 code.

1 Q. They don't have any impact, whatsoever, in how the program
2 runs once it's compiled?

3 A. That's correct.

4 Q. And you don't know whether either of these files, that
5 these were in whatever part of a handset?

6 A. I don't recall that, no.

7 Q. Now, the other eight files, those are -- all have the word
8 "Impl" on it. That's part of the name, the API name; is that
9 right?

10 A. I believe it's some or perhaps all of them.

11 Q. And these all appear to have in the name header the word
12 "Test," correct?

13 A. No.

14 MR. VAN NEST: Let's open up page 8 of Dr. Visnick's
15 report, please, and display that. Let's go to the next page.
16 It's the one with all the code files on it. It's paragraph 22.
17 I'm sorry. There we go.

18 (Document displayed.)

19 BY MR. VAN NEST:

20 Q. And are the eight Impl files displayed there on the
21 left-hand side of the page, starting in paragraph 22?

22 A. Looking at the table below paragraph 22?

23 Q. Yes, sir.

24 A. I think the table shows the Android file name, the Sun
25 Java file name, and then some description.

1 Q. The Android name has the word "test" in it, right?

2 A. The Android file path leading to this file contains the
3 word "test."

4 MR. VAN NEST: And could we highlight the first line
5 there in the second box, Ben, so jurors can follow us here.
6 There it is. Okay.

7 (Document displayed.)

8 BY MR. VAN NEST:

9 Q. The Android files on the left, Java files on the right,
10 the Android file has the word "test" in it, correct?

11 A. Yes. The Java name does not.

12 Q. And all eight, all eight of these files that you've
13 identified here, they have all the word "test"?

14 A. That's easy to check. I think that's true.

15 Q. And you don't actually know whether any of these files was
16 ever compiled and placed onto a handset; do you?

17 A. I don't believe I have a reason to believe that they were.

18 Q. So you don't have any reason to think that they were ever
19 put on a handset, right?

20 A. I don't know one way or the other.

21 Q. All right. So in terms of your code-by-code or
22 code-to-code line-to-line analysis, the only thing you're aware
23 of that you can say for sure ever made it onto a phone was
24 these nine lines of code in rangeCheck, right?

25 A. Of the things you're -- of those 12 files, or from what

1 larger class?

2 Q. Of the 12 files that Dr. Visnick identified as the only
3 files in Android as to which he could say he thought they were
4 copied. That's what I mean.

5 A. I believe that's correct, that, of these, the ones I know
6 appear on the phone are from the Timsort files.

7 Q. So, again, the only thing that you've identified in this
8 exhaustive analysis was nine lines of code that made it onto
9 the handset. Right, Dr. Mitchell?

10 A. That's not the only thing identified. Is that your
11 question?

12 Q. Out of these 12 files, the only --

13 A. There are lots of things identified here. Could you
14 just -- I'm not sure I understand your question.

15 Q. The 12 files that you and Dr. Visnick identified and you
16 testified were copied code to code, only nine lines of that
17 code ever made it onto a handset?

18 A. That may be true.

19 Q. Now, Dr. Visnick also ran his analysis against the Apache
20 Harmony code, right?

21 A. If his report says so, yes.

22 Q. You know what Apache Harmony is?

23 A. I'm vaguely familiar with that.

24 Q. That's an independent implementation of Java libraries?

25 MR. JACOBS: Objection, Your Honor. Formulation.

1 "Independent implementation." The witness has no knowledge of
2 how Apache was created.

3 **THE COURT:** Do you have any knowledge -- not
4 knowledge, but from your experience and training do you have an
5 opinion on that subject?

6 **THE WITNESS:** I really haven't looked into Apache
7 Harmony.

8 **THE COURT:** All right. Well, then, ask the question.
9 The witness cannot answer the question as phrased. You need to
10 ask it a different way.

11 **BY MR. VAN NEST:**

12 **Q.** Okay. Let's -- you do know, you do know that one of the
13 things that Dr. Visnick analyzed, in addition to Android, was
14 the Apache Harmony code, right?

15 **A.** Uhm, again, if that's what his report says, then I
16 believe --

17 **Q.** Let's go to paragraph 10. Paragraph 10.

18 **MR. JACOBS:** Your Honor, we've allowed Mr. Van Nest
19 some latitude here, but he's now just asking this witness to
20 repeat what another witness put in a report that is not part of
21 Dr. Mitchell's testimony.

22 **THE COURT:** Well, but, this witness relied on that
23 report.

24 **MR. JACOBS:** He was very clear on what he did with
25 that report, Your Honor. He took the particular files that

1 Dr. Visnick identified, and then studied them himself.

2 He has testified to nothing about copying Apache
3 Harmony. He hasn't said anything in his report or in his
4 testimony about whether Apache Harmony was copied or not. And
5 now we're just asking this witness to repeat what's in
6 Dr. Visnick's report, whether or not he relied on it.

7 Let me go backwards. He hasn't established that this
8 witness had any -- made any reliance on the portion of the
9 report that Mr. Van Nest is now asking about.

10 **THE COURT:** Well, possibly. But, Mr. Van Nest, you
11 may ask the witness did he read the entire report, and why did
12 he read it, for what purpose did he read it. And then we'll
13 see where we are. Go ahead.

14 **BY MR. VAN NEST:**

15 **Q.** I think you testified previously, Dr. Mitchell, that you
16 relied on Dr. Visnick's report. Did I have that right?

17 **A.** I don't remember testifying to that, but I may have if the
18 record is clear.

19 **THE COURT:** Wait a minute. You didn't do the
20 homework of looking up these files yourself, right?

21 **THE WITNESS:** That's correct.

22 **THE COURT:** You relied on Mr. Visnick, and then you
23 took the one that he found, and you confirmed it. That part I
24 understood. Right?

25 **THE WITNESS:** That's correct.

1 **THE COURT:** All right. So you relied on his report
2 to give you a heads up as to which ones he thought were word
3 for word or line for line, or whatever that phrase was, right?

4 **THE WITNESS:** Yes.

5 **THE COURT:** Okay. So if he had had a hundred files
6 in there, you would have read a hundred? You would have done a
7 hundred. But he only had 12. That's what I'm getting out of
8 this. Right?

9 **THE WITNESS:** Absolutely.

10 **THE COURT:** So, now, did you read the part about
11 Apache Harmony, too?

12 **THE WITNESS:** I probably read it. I wasn't asked to
13 look into Apache Harmony, and didn't.

14 **THE COURT:** All right. We're going to skip Apache
15 Harmony, and you can put your own witness on about that. I
16 don't see the point of getting into Apache Harmony with this
17 witness.

18 **MR. VAN NEST:** Fair enough, Your Honor.

19 All right. Dr. Mitchell.

20 **BY MR. VAN NEST:**

21 **Q.** All right. Dr. Mitchell, there are, as part of the APIs,
22 descriptions of what each method and what each class does,
23 correct?

24 **A.** Yeah. The API contains English text that describes the
25 way each method works.

1 Q. And that text we sometimes call the specification?

2 A. I think the whole thing is a specification. "API" and
3 "API specification" are often used synonymously.

4 Q. So the text descriptions, they describe what the methods
5 or the classes or the packages do?

6 A. Yeah. They give meaning to the declarations that are in
7 the API.

8 Q. And they're sort of -- would it be fair to refer to those,
9 sort of a users-manual-type thing?

10 A. I don't think that's really accurate. They serve two
11 purposes. They are helpful for users, programmers that use the
12 library. And they are also used by implementers, if the API is
13 designed and to be later implemented.

14 Q. And there are thousands and thousands of those
15 descriptions in Java and in Android, as well?

16 A. I think there are 5,000 methods in the 35 packages that I
17 looked at. So there are thousands and thousands, certainly,
18 correct.

19 Q. And there's one for every method?

20 A. Methods are all described, as far as I know.

21 Q. One for every class?

22 A. Yes.

23 Q. One for every package?

24 A. Yes.

25 Q. And those are all written in English text so anybody can

1 read them?

2 **A.** Anybody who reads English, yes.

3 **Q.** And you didn't bother to provide any examples of those
4 during your direct examination; did you?

5 **A.** Yes, I had some slides that showed some examples. And I
6 discussed them.

7 **Q.** Is it your testimony that all of the thousands of lines of
8 descriptions in Android are just exactly the same as in Java?

9 **A.** No.

10 **Q.** There are many, many, many, many differences in the
11 English-text descriptions in the Android version as contrasted
12 with the Java version. Isn't that right, Dr. Mitchell?

13 **A.** Based on my study, I believe they say basically the same
14 thing, but the wording can differ.

15 **Q.** And the wording does differ, right?

16 **A.** Yes. Sometimes it's the same. Sometimes it's different.

17 **Q.** And, actually, you put -- you put exactly one example of
18 this in your report, correct?

19 **A.** I think there are a few exhibits that have examples of
20 code side by side. In my opening report it may have discussed
21 one, primarily.

22 **Q.** So the one you chose to discuss was in paragraph 207.

23 **MR. VAN NEST:** Could we have 207 up please.

24 (Document displayed.)

25

1 **BY MR. VAN NEST:**

2 **Q.** This is the only paragraph -- your report has several
3 hundred paragraphs, right?

4 **A.** Yes.

5 **Q.** Almost 280 paragraphs. This is the one paragraph that has
6 anything in it comparing the English-text descriptions in
7 Android and in Java, right?

8 **A.** Yes. Also, as you can see in the paragraph, it refers to
9 exhibits that show code.

10 **Q.** And the example you chose here is a method called
11 `KeyPair.getPrivate`, right?

12 **A.** That's the example considered in this paragraph, yes.

13 **MR. VAN NEST:** Could we put up the Java version,
14 then, on top or to the left. Okay.

15 **BY MR. VAN NEST:**

16 **Q.** Now, `KeyPair`, that's a method like `max` and `square root` and
17 these other methods we've been talking about, right?

18 **A.** Yes.

19 **Q.** Dr. Mitchell, `KeyPair`, all I'm asking is, `KeyPair` is a
20 method?

21 **A.** I believe so. I don't think I have the exhibits here, so
22 I can't see the code.

23 **Q.** You don't have your report?

24 **A.** I have the report.

25 **Q.** It's paragraph 207. All I'm doing -- the computer is

1 pulling the language out so we can see it better.

2 **A.** Okay. Fine.

3 **Q.** So KeyPair wasn't invented by Sun. KeyPair has been
4 around, right?

5 **A.** Well, in cryptography, in public-key cryptography, the
6 cryptographic keys come in pairs.

7 **Q.** That's not something Sun invented; that's been around a
8 long time?

9 **A.** That concept comes from public-key cryptography, which was
10 invented in the late 1970s.

11 **Q.** So that's been here for a long, long time, right?

12 **A.** Yes.

13 **Q.** Yeah.

14 Now, what -- what we're showing here on the left,
15 that is the description, the text description in Java and
16 KeyPair, right? Returns a reference to the PrivateKey
17 component of this key pair, right?

18 **A.** Yes.

19 **MR. VAN NEST:** And can we show the Android version on
20 the right.

21 (Document displayed.)

22 **BY MR. VAN NEST:**

23 **Q.** Is that it, returns the private key?

24 **A.** I believe so.

25 **Q.** Now, does that pass your test for being substantially

1 similar?

2 **A.** I think if you were considering using this method, both
3 would tell you the same information, that you can use this
4 method to get the private key component of the key pair.

5 **Q.** That was not my question. That was not my question. My
6 question is:

7 Does this pass your test for being substantially
8 similar descriptions of the very same thing?

9 **A.** I think in the context of the rest of the description they
10 mean the same thing.

11 **Q.** Again, that wasn't my question.

12 They mean the same thing. They're both describing
13 the same thing, right? They're both describing the KeyPair
14 method. Right, Dr. Mitchell?

15 **A.** They both give the same meaning to that. If you were to
16 read it separately in the Android spec, you would conclude that
17 it was the same method in the sense of performing the same
18 operations and returning the same result.

19 **Q.** My question is whether applying your test of
20 "substantially similar" the phrase "returns a reference to the
21 private key component of this key pair" is substantially
22 similar to "returns the private key"?

23 **THE COURT:** You can answer yes or no.

24 **THE WITNESS:** Yes.

25 **THE COURT:** Thank you. Next question.

1 **MR. VAN NEST:** I have nothing further, Your Honor.

2 **THE COURT:** All right. Let's go to redirect.

3 **REDIRECT EXAMINATION**

4 **BY MR. JACOBS:**

5 **Q.** Dr. Mitchell, Google's counsel asked you about the
6 independent economic significance of rangeCheck. Do you recall
7 that question?

8 **A.** Yes.

9 **Q.** Did you conduct an analysis of the significance of
10 rangeCheck to other code in the same class file?

11 **A.** Yes.

12 **Q.** What did you conclude?

13 **A.** I found a number of other source code in other files that
14 called that function.

15 And, also, I did an experiment with the phone source
16 code instrumented, and counted the number of times that
17 rangeCheck was called in starting up the phone. And I found
18 that it's called 2600 times just in powering on the device or
19 starting the emulator.

20 So 2600 seems like a pretty big number for the number
21 of calls to this function.

22 **Q.** And then on the decompiled code, did you analyze the
23 significance of the decompiled code?

24 **A.** I looked into what the purpose of that set of source code
25 files is. They have to do with access control lists, which are

1 a standard mechanism in computer security to govern access to a
2 file or a network or other resource.

3 An access control list lists the individuals, the
4 users that are allowed to read a file or present opposite kinds
5 of information in various ways.

6 Q. Mr. Van Nest asked you about whether they were test files
7 or not. Do you recall that questioning?

8 A. Yes, I do.

9 Q. And on the Java side, are these test files?

10 A. I don't believe so. They are characterized as the default
11 implementation for the security functions.

12 Q. And on the Android side, are they test files?

13 A. I don't know. They are in a directory that has the word
14 "test," but they could be used in other ways besides testing.

15 Q. All right. I want you to assume that they are test files
16 in Android. Does that mean they are unimportant?

17 A. I think in the software development process, code is
18 written and then tested. And by many measures the testing and
19 quality assurance process can be twice as time consuming or
20 twice as expensive as coding originally.

21 So testing is a very important part of the software
22 development. It's expensive. And software companies want to
23 do it correctly so that the code that they ship is bug free and
24 usable to their customers.

25 Q. So what advantage would the Google developers have

1 obtained by going through this exercise of decompiling Java
2 object code in order to copy it into Android?

3 **A.** Well, if this helped them test other code they were
4 developing, and speed up and lessen the cost of testing and
5 quality assurance, then that would have a big value to them.

6 **Q.** Separating out the code-to-code examples. Google's
7 counsel was also very precise in distinguishing the
8 code-to-code copying from the API implementations, so I want to
9 set aside the code-to-code copying for a minutes.

10 What is the significance of the decision in Android
11 to copy the 37 API packages into the Android platform?

12 **A.** Oh, so, that's huge. These 37 packages are highly useful.
13 They're important to developers and programmers. And to have
14 the same API as the Oracle Java library has a very high
15 significance to any potential developers.

16 **Q.** In the way that Google took the 37 packages, did they make
17 Android compatible, quote/unquote, compatible with Java?

18 **A.** Not really, because they subsetting some of these. And
19 they also superseded by adding other packages and changing the
20 set of libraries that an Android developer might use.

21 **Q.** And so what does that mean, if a Java application is
22 installed on an Android phone? Will it run or not?

23 **A.** So a Java application that was written with the Sun/Oracle
24 library in mind, won't run on Android if it relies on things
25 that weren't there.

1 So you don't really have compatibility. You can't
2 ship code from one platform to another.

3 **Q.** Google's counsel asked you some questions about the
4 relationship between the APIs and the language. And he applied
5 the test of the back cover of the programming language book.
6 Do you recall that?

7 **A.** Yes.

8 **Q.** And then you said -- and I wanted to give you a chance to
9 expand on your answer -- that there are two other ways to
10 analyze what constructs from the application programming
11 interfaces are actually properly thought of when you're talking
12 about the Java programming language.

13 **A.** Right. So --

14 **Q.** What did you mean by that?

15 **A.** Yeah. So two methods that I think make sense and lead to
16 a conclusion are, one, treat the language definition as a
17 definition of what's in the language and what isn't.

18 Mark Reinhold, Dr. Mark Reinhold -- who is the chief
19 architect of the platform and is very knowledgeable about
20 that -- carried out a careful study and came up with 60 or 61
21 classes that by that criterion can be considered part of the
22 language.

23 Another way to figure out what's part of the language
24 is to look at the language compiler. The language compiler
25 takes source code written in Java and compiles it to bytecode

1 so it can run.

2 If the compiler knows about some part of some source
3 code and can compile it directly, then I think it makes sense
4 to consider that part of the language.

5 If the compiler doesn't know what some language
6 construct means, and it has to look up prewritten program to
7 understand how to run that program, then I think it's accurate
8 or fair to consider that not part of the language.

9 **Q.** And so based on those analyses, what is your conclusion
10 about what elements of what classes are required by the Java
11 programming language? Is it the same as or different from
12 Dr. Reinhold's?

13 **A.** I like his analysis. I think it was very thorough. These
14 two criteria seem very sensible to me.

15 I think the compiler study came up with slightly
16 fewer classes. So those 60 or 61, whatever his number is,
17 seems like a very good answer. And the compiler answer seems
18 consistent with that, and also a good answer to the question.

19 **Q.** Thank you.

20 I need you to do one more thing so that our record is
21 absolutely clear. I need you to turn to your report, and I
22 need you to tell us the names of the 37 packages that are
23 accused of infringement based on their copying of the Java API
24 specification.

25 **THE COURT:** Well, may I -- do we have a handwritten

1 list somewhere of this?

2 **MR. JACOBS:** We do, Your Honor.

3 **THE COURT:** I'm going to let you show the handwritten
4 list. And we'll put that in evidence as long as that's all it
5 is.

6 Any objection to proceeding in that manner?

7 **MR. VAN NEST:** No, Your Honor.

8 **THE COURT:** Otherwise, we are going to be taxing the
9 abilities of the jury and the judge and the court reporter to
10 get it exactly down right.

11 So I want the witness to look at it, Counsel, to look
12 at it. And then if it's in good order, we're going to receive
13 that in evidence, and the witness can testify to what it is.

14 **MR. JACOBS:** I gave it to counsel before the break.
15 And so I'll hand a copy to the witness now.

16 **THE COURT:** Is that the Android version or the Java
17 version?

18 **MR. JACOBS:** Well, it would be the same names, but it
19 is "Accused API Packages and Files in Android." It is trial
20 Exhibit 1072. It includes the names of both the API packages
21 and the files. And if the witness agrees with it, I would move
22 it into evidence.

23 **THE WITNESS:** Yes, I'm just comparing it with the
24 list in my report.

25 It seems to be the same list of packages. The files

1 listed on this page are in -- can be divided into three
2 categories. The three categories are indicated on the foot
3 chart. And I think these are the correct file names.

4 **THE COURT:** Any objection to 1072?

5 **MR. VAN NEST:** No, Your Honor.

6 **THE COURT:** Received in evidence.

7 (Trial Exhibit 1072 received in evidence.)

8 **BY MR. JACOBS:**

9 **Q.** Dr. Mitchell, if all that Google wanted to do was
10 implement the Java programming language, as you described it a
11 few minutes ago, would they have needed to copy the API
12 packages, any of the API packages wholesale, those packages
13 that are listed on 1072?

14 **MR. VAN NEST:** Objection, Your Honor. Calls for
15 speculation. It's argument.

16 **MR. JACOBS:** It's fairly within the scope of the
17 questions he was asked on --

18 **THE COURT:** It is within the scope of the type of
19 questions that you asked on cross, so this is a permissible
20 question.

21 So, please, answer.

22 **THE WITNESS:** The Java language could be implemented
23 using, at most, the 60 classes. And none -- those 60 classes
24 don't -- wouldn't use -- require all or any of these packages.

25 **MR. JACOBS:** Thank you very much, Dr. Mitchell. No

1 further questions.

2 **THE COURT:** Anything more?

3 **MR. VAN NEST:** I have nothing further, Your Honor.

4 **THE COURT:** All right. Dr. Mitchell, you may -- I
5 guess he's possibly coming back on a reply report, right --

6 **MR. JACOBS:** Yes, Your Honor.

7 **THE COURT:** -- at some point.

8 So we'll possibly see you again later on in the
9 trial. But thank you for now. Leave all of our documents
10 here.

11 **THE WITNESS:** All yours. Thank you.

12 **THE COURT:** And you may step down. Thank you.

13 **THE WITNESS:** Thank you.

14 (Witness steps down.)

15 **THE COURT:** All right. So we have enough time.
16 We're going to start the next witness.

17 **MR. BOIES:** Your Honor, we call as our next witness,
18 Mr. Rubin.

19 **THE COURT:** Very well. Let's bring him in.

20 **MR. JACOBS:** While we're doing that, Your Honor,
21 could I ask you to read a -- one of the deemed admitted
22 stipulations to the jury?

23 **THE COURT:** I would be happy to. You're going to
24 have to hand it up to me, though. Show the counsel which one
25 you want.

1 (Pause)

2 **THE COURT:** Okay. As the witness is coming
3 forward -- let's have him come up.

4 Counsel would you clear away the witness stand so
5 that he'll have a clear deck.

6 Mr. Rubin, welcome.

7 **THE WITNESS:** Thank you.

8 **THE COURT:** Please just sit right there. We need to
9 do something here.

10 Which one of these do you want me to read?

11 **MR. JACOBS:** Right in the middle there, Your Honor,
12 "The 37 packages."

13 **THE COURT:** All right. Okay. You remember I said
14 that there were some items that I would give to you that will
15 be deemed to be evidence in the case. It's not conclusive.
16 It's not, you know, like written in gold letters or something.
17 It's just evidence that counts as much as any other evidence in
18 the case.

19 So, here we go. I'll do this very slowly because
20 this is probably the only time you'll hear it.

21 "For the 37 accused API packages, Android and
22 Java 2 SE version 5.0 have substantially the
23 same selection, arrangement, and structure of
24 API elements."

25 That's the one you wanted me to read?

1 **MR. JACOBS:** Yes, Your Honor.

2 **THE COURT:** All right. I'll read it again, this time
3 a little faster.

4 "For the 37 accused API packages, Android and
5 Java 2 SE version 5.0 have substantially the
6 same selection, arrangement, and structure of
7 API elements." Period. End of item.

8 Thank you.

9 We will now go to the next witness. Mr. Rubin,
10 please stand and raise your right hand.

11 **ANDY RUBIN,**

12 called as a witness for the Plaintiff herein, having been first
13 duly sworn, was examined and testified as follows:

14 **THE WITNESS:** Yes, I do.

15 **THE CLERK:** Okay. Thank you.

16 **THE COURT:** Welcome, again.

17 Please, you have to sit about this close to the mic.
18 And the mic, though, will move around to suit your convenience.
19 It will pull back. Why don't you pull it back a little. It --
20 no, the other way. Towards you.

21 **THE WITNESS:** Towards me. I see.

22 **THE COURT:** You need to be about this close. Why
23 don't you say your name.

24 **THE WITNESS:** Andy Rubin.

25 **THE COURT:** All right. Go ahead, counsel.

1 **MR. BOIES:** Thank you, Your Honor.

2 **DIRECT EXAMINATION**

3 **BY MR. BOIES:**

4 **Q.** Good afternoon, Mr. Rubin.

5 **A.** Good afternoon.

6 **Q.** You are, and have been since 2005, the Google executive in
7 charge of Android, correct?

8 **A.** That's correct.

9 **Q.** And before joining Google in 2005, you had been employed
10 by Android, Incorporated?

11 **A.** That's correct.

12 **Q.** And before that you had been employed by Danger?

13 **A.** That's correct.

14 **Q.** Now --

15 **A.** Actually, can I go back a little bit? In between Danger
16 and Android, I was employed by a company called Red Point
17 Ventures.

18 **Q.** Now, there have been some words that I want to be sure I
19 have the same use of those words as you do, that have been
20 talked about in this trial.

21 One has been "Java core APIs." Is that a term you're
22 familiar with?

23 **A.** I would need to -- I have my interpretation of that term,
24 yes.

25 **Q.** And what is your interpretation of that?

1 A. Uhm, that it is a term that is often used by Sun
2 Microsystems and Oracle to describe part of their product.

3 Q. Which part of the product?

4 A. The part of the product that involves the Java language
5 class files.

6 Q. And how many APIs are those?

7 A. I don't know.

8 Q. Approximately?

9 A. I don't know.

10 Q. Did you look at that at all in the course of the many
11 years you've been working on Android?

12 A. Yeah, I didn't count them though, but I looked at them.

13 Q. I'm not asking for an exact count but just approximately.
14 Could you give us that?

15 A. I -- I'm sorry, I just don't have a -- you know, Sun
16 represents that they have core APIs. I don't know how many
17 core APIs they have.

18 Q. Did you -- do you, in your own language, use core APIs to
19 mean the same thing as core libraries?

20 A. Uhm, I don't use the term "core APIs" or "core libraries."

21 Q. You don't use the word "core libraries"?

22 A. No.

23 Q. How about "core packages," do you use that term?

24 A. Uhm, I've used core packages before, I believe, yes.

25 Q. And when you use core packages, what are you referring to?

1 **A.** It depends in what context I used it. I think it can be
2 used in various contexts, but I'm not sure what you're
3 referring to.

4 **Q.** Well, when you use the term "core packages" in your
5 business, what do you mean by it?

6 **A.** Typically, I'm describing a set of -- a set of methods and
7 packages that -- that an application developer would develop
8 for.

9 **Q.** Now, I'm going to show you some documents that either you
10 participated in preparing or that you received in the course of
11 your work. And I'm going to try to get these moved into
12 evidence as quick as possible.

13 **MR. BOIES:** May I approach, Your Honor?

14 **THE COURT:** You may.

15 **MR. BOIES:** These are in folders in numerical order.
16 The first one is Trial Exhibit 14. And I would offer that as a
17 party admission.

18 **THE COURT:** Any objection?

19 **MR. VAN NEST:** No objection, Your Honor.

20 **THE COURT:** Number 14 is in evidence.

21 (Trial Exhibit 14 received in evidence.)

22 **MR. BOIES:** The next one is Trial Exhibit 15, which I
23 would offer as a party admission.

24 **MR. VAN NEST:** No objection, Your Honor.

25 **THE COURT:** Thank you. Received.

1 (Trial Exhibit 15 received in evidence.)

2 **MR. BOIES:** I would offer Trial Exhibit 18 as a party
3 admission.

4 **THE COURT:** Any objection?

5 **MR. VAN NEST:** I'm not sure what counsel means by
6 "party admission."

7 **THE COURT:** It's the same problem we had the other
8 day. I don't want the jury to think that somehow whenever I
9 said it's admitted that you over there in the jury box think,
10 aha, this is a party admission, this ends the case. It's like
11 a guilty plea. No. That's not what counsel means. Just so --
12 it's an evidentiary phrase.

13 So, Mr. Boies, I think it's better just to say you
14 offer Exhibit 18. And if there's an objection, then we'll get
15 into an argument over it. But I don't want the jury to think
16 that when I admit it, it's somehow -- all of these documents,
17 more or less, are from one side's files or the other, so --
18 okay.

19 What's next?

20 **MR. BOIES:** Offer Exhibit 18.

21 **THE COURT:** Received. Next.

22 (Trial Exhibit 18 received in evidence.)

23 **MR. BOIES:** Offer Exhibit 22.

24 **THE COURT:** Received.

25 (Trial Exhibit 22 received in evidence.)

1 **MR. BOIES:** Offer Exhibit 29.

2 **THE COURT:** Received.

3 (Trial Exhibit 29 received in evidence.)

4 **MR. VAN NEST:** No objection, Your Honor.

5 **MR. BOIES:** Offer Exhibit 131.

6 **MR. VAN NEST:** No objection, Your Honor.

7 **THE COURT:** Received.

8 (Trial Exhibit 131 received in evidence.)

9 **MR. BOIES:** Offer Exhibit 140.

10 **MR. VAN NEST:** No objection, Your Honor.

11 **THE COURT:** Received.

12 (Trial Exhibit 140 received in evidence.)

13 **MR. BOIES:** Offer Exhibit 147.

14 **MR. VAN NEST:** No objection, Your Honor.

15 **THE COURT:** All right. All received.

16 (Trial Exhibit 147 received in evidence.)

17 **MR. BOIES:** Offer Exhibit 151.

18 **MR. VAN NEST:** No objection.

19 **THE COURT:** Received.

20 (Trial Exhibit 151 received in evidence.)

21 **THE COURT:** Thank you. And?

22 **MR. BOIES:** Offer Exhibit 154.

23 **MR. VAN NEST:** No objection, Your Honor.

24 **THE COURT:** Received.

25 (Trial Exhibit 154 received in evidence.)

1 **MR. BOIES:** Offer Exhibit 158.

2 **MR. VAN NEST:** No objection, Your Honor.

3 **THE COURT:** Received.

4 (Trial Exhibit 158 received in evidence.)

5 **MR. BOIES:** Offer Exhibit 161.

6 **MR. VAN NEST:** No objection, Your Honor.

7 **THE COURT:** Received.

8 (Trial Exhibit 161 received in evidence.)

9 **MR. BOIES:** Offer Exhibit 165.

10 **MR. VAN NEST:** No objection, Your Honor.

11 **THE COURT:** Received.

12 (Trial Exhibit 165 received in evidence.)

13 **MR. BOIES:** Offer Exhibit 180.

14 **THE COURT:** Received. Any objection?

15 **MR. VAN NEST:** No objection.

16 **THE COURT:** Thank you. Received.

17 (Trial Exhibit 180 received in evidence.)

18 **MR. BOIES:** Offer Exhibit 183.

19 **MR. VAN NEST:** No objection.

20 **THE COURT:** And?

21 **MR. BOIES:** Offer Exhibit 203.

22 **MR. VAN NEST:** Which is the number? Excuse me.

23 **MR. BOIES:** 203.

24 **MR. VAN NEST:** No objection.

25 **THE COURT:** Received.

1 (Trial Exhibits 183 and 203 received in evidence.)

2 **THE COURT:** How many more of these do you have?

3 **MR. BOIES:** Probably another 30, Your Honor.

4 **THE COURT:** We'll do that later. I think while the
5 jury is here, we'll done do this offline. I want you to start
6 your examination so we'll take advantage -- unless you need
7 some of these for the examination, we only have 15 minutes to
8 go.

9 So if you -- if you get to one of these that's not
10 yet in, we'll allow you to move that one in. Otherwise, we'll
11 take these items out of the presence of the jury.

12 Go ahead.

13 **MR. BOIES:** Thank you, Your Honor.

14 Let me ask that the witness look at Trial Exhibit 3
15 that's already in evidence.

16 May I approach?

17 **THE COURT:** Yes.

18 **BY MR. BOIES:**

19 **Q.** And you're familiar with this document, are you not, sir?

20 **A.** Uhm, I have a faint recollection, yes.

21 **Q.** This is a document, the first page of which is an e-mail
22 from you to Mr. Lindholm, correct?

23 **A.** Yes, that's correct.

24 **Q.** And you wrote this July 29th, 2005, correct?

25 **A.** Yes.

1 Q. And on the third page of the exhibit, under the heading of
2 requirements, do you see where you write "Google needs a TCK
3 license"?

4 A. Yes, I see that.

5 Q. Now, Google did not need a license for the Java
6 programming language, correct?

7 A. That's what I believe correct.

8 Q. Uhm, so when you're talking about needing a TCK license,
9 you're not talking about a license for the Java programming
10 language, correct, sir?

11 A. That's correct.

12 Q. Now, let me ask you to look next at Trial Exhibit 7. And
13 you're familiar with this document, are you not, sir?

14 A. Yes, I am.

15 Q. And part of this document is an e-mail that you wrote to
16 Mr. Larry Page on October 11, 2005, correct, sir?

17 A. Yes.

18 Q. And in the second sentence you say -- second paragraph:

19 "My proposal is that we take a license that
20 specifically grants the right for us to open
21 source our product."

22 And that's a license from Sun, correct, sir?

23 A. That's correct.

24 Q. And you say:

25 "We'll pay Sun for the license and the TCK."

1 Do you see that?

2 A. Yes, I do.

3 Q. And, again, the license that you're talking about there is
4 not a license for the Java programming language, correct?

5 A. That's correct.

6 Q. You then go on to say:

7 "Before we release our product to the open
8 source community, we'll make sure our JVM
9 passes all TCK certification tests so that we
10 don't create fragmentation."

11 Do you see that?

12 A. Yes, I do.

13 Q. And JVM there stands for Java virtual machine?

14 A. That's correct.

15 Q. And then you go on to say:

16 "Before a product gets brought to market, a
17 manufacturer will have to be a Sun licensee,
18 pay appropriate royalties and pass the TCK
19 again."

20 Do you see that?

21 A. Yes, I see that.

22 Q. And when you talk about a manufacturer will have to be a
23 Sun licensee, you're talking about the manufacturer taking a
24 license from Sun, correct?

25 A. Well, I'm -- I assume that a lot of the manufacturers

1 already had licenses with Sun.

2 Q. That is they would either already have to have a license
3 or, if they didn't have a license, they'd have to take one --

4 A. Yes.

5 Q. -- correct?

6 A. Uh-huh.

7 Q. Now, let me ask you to look next at Exhibit 12. And this
8 also is a document that you're familiar with, correct, sir?

9 A. Yes, I'm familiar with it.

10 Q. And this is a couple of e-mails. One of them is a e-mail
11 that you wrote, correct, sir?

12 A. In the middle there, yes, I see it.

13 Q. And you say:

14 "My reasoning is that either a) we'll partner
15 with Sun as contemplated in our recent
16 discussions or b) we'll take a license."

17 Do you see that?

18 A. Yes, I see that.

19 Q. And you were, at this point in time, which is December of
20 2005, the end of December, you were having discussions about
21 possibly partnering with Sun, correct?

22 A. We were in discussions with Sun for quite some time.
23 Partnership was one of my main objectives, yes.

24 Q. In fact, what you say, "We'll partner with Sun as
25 contemplated in our recent discussions," correct, sir?

1 A. Yes, I was in recent discussions with Sun at the time I
2 wrote this e-mail.

3 Q. And if those discussions didn't work out, then you were
4 going to take a license from Sun, correct?

5 A. For some technology from Sun, yes.

6 Q. What you say here is, "We'll take a license," correct?

7 A. Those were the options. Those were a couple of the
8 options at the time.

9 Q. You didn't say those were a couple of the options. You
10 said, my reasoning is that either, a, we'll do the partnership;
11 or, b, we'll take a license.

12 That's what you said, right, sir?

13 A. That was my reasoning, yes.

14 Q. And you then go on to say:

15 "I think a clean-room implementation is
16 unlikely because of the team's prior
17 knowledge."

18 Do you see that?

19 A. Yes, I do.

20 Q. And what you're talking about there, the team you're
21 talking about is the Google team that's developing Android,
22 correct?

23 A. I wasn't specific about which team I was talking about
24 there.

25 Q. Well, sir, what team were you talking about?

1 A. It would have been any team developing the clean-room
2 implementation at the time.

3 Q. Clean-room implementation of what?

4 A. Clean-room implementation of a Java virtual machine.

5 Q. For Android, correct?

6 A. Yes.

7 Q. Okay. And what you're saying is that the Google team
8 responsible -- or teams responsible for developing the Java
9 virtual machine for Android had too much prior knowledge to
10 operate in a clean room environment, right, sir?

11 A. I think that's reading a lot into that small sentence. I
12 wouldn't go that far, actually.

13 Q. Well, let's read that small sentence.

14 "I think a clean-room implementation is unlikely."

15 And you're talking there about a clean-room
16 implementation of the Java virtual machine that you're
17 developing for Android, correct, sir?

18 A. That we were contemplating developing at this time.

19 Q. And that you did develop, right?

20 A. Yeah, but at this time it hadn't -- I'm unsure whether it
21 had begun or not.

22 Q. Excuse me, sir?

23 A. At this time I'm unsure whether it had begun or not.

24 Q. But you certainly were planning to begin it if you hadn't
25 begun it; that's fair?

1 A. We hadn't actually made a decision to implement a
2 clean-room VM at this point.

3 Q. Right at the very beginning, back in July when you made
4 the first GPS presentation, you were talking about a clean-room
5 implementation, weren't you, sir?

6 A. We were talking about strategies in general of how to
7 accomplish our goals.

8 Q. I'm sure that's true. But in particular, you were talking
9 about having a clean-room implementation, correct?

10 A. We were, if I recall correctly -- and I'm vague on that
11 actual presentation, but --

12 Q. All right. Let me give it to you, sir.

13 A. Thank you.

14 Q. This is Trial Exhibit 1. It's in evidence.

15 (Document displayed.)

16 This is the Android GPS presentation, July 26, 2005;
17 correct, sir?

18 A. Yes.

19 Q. And you made this presentation; did you not, sir?

20 A. I did.

21 Q. And you made it to the very top executives of Google; did
22 you not, sir?

23 A. I presented it, yes, to our executive staff.

24 Q. And let me ask you to turn to page 9 of the exhibit, where
25 it has a big heading of "JAVA."

1 A. Okay.

2 Q. And do you see where it says:

3 "Current scenario, developing a clean-room
4 implementation of a Java virtual machine"?

5 A. Yes.

6 Q. And now go back to Exhibit 12.

7 (Document displayed.)

8 When you talk about, in Exhibit 12, in December,
9 December 20, 2005, and you say, "I think a clean-room
10 implementation is unlikely," you're talking about a clean-room
11 implementation of a Java virtual machine for Android, correct,
12 sir?

13 A. Yes.

14 Q. Okay. And then you go on and give the reason why you
15 think a clean-room implementation of a Java virtual machine for
16 Android is unlikely. Do you not, sir?

17 A. I give a reason, yes.

18 Q. And the reason is because of the "teams prior knowledge,
19 and" -- second reason -- "it would be uncharacteristically
20 aggressive of us to position ourselves against the industry,"
21 correct, sir?

22 A. Yes.

23 Q. Now, let me ask you to look next at Exhibits 17 and 18.

24 First Exhibit 17.

25 A. Thank you.

1 Q. First, with respect to Exhibit 17 -- this is Trial Exhibit
2 17, and it is dated February 10, 2006, correct, sir?

3 A. Yes.

4 Q. And there's a e-mail here from Mr. Lindholm to
5 Mr. Coughran and to you, correct?

6 A. Yes.

7 Q. And what Mr. Lindholm's says at the very beginning is:

8 "I have been helping Andy Rubin with some
9 issues associated with his Android platform.

10 This has mostly taken the form of helping
11 negotiate with my old team at Sun for a
12 critical license."

13 Do you see that?

14 A. Yes, I do.

15 Q. And, again, we're in agreement that that was not a license
16 for the Java programming language because you didn't need a
17 license for the Java programming language, correct, sir?

18 A. That's correct.

19 Q. Now let me ask you to look at 18.

20 (Document displayed.)

21 MR. BOIES: I'm not sure whether I got as far as 18
22 when I was offering exhibits.

23 THE COURT: 18 is in evidence.

24 MR. BOIES: Thank you, Your Honor.

25

1 BY MR. BOIES:

2 Q. And there is, in the middle of the page, a e-mail that you
3 wrote on March 24th, 2006, correct, sir?

4 A. Uhm, I see that, yes.

5 Q. And that was in response to an e-mail that you had
6 received earlier that day from Greg Stein, correct?

7 A. Uhm, it's hard to say whether it was earlier, but it was
8 on the same date.

9 Q. Well, you're writing in response to it, right?

10 A. Yeah. I'm just don't want to confuse the ordering of
11 this, because it doesn't have a time on my message.

12 Q. But if you're writing in response, it would have had to be
13 something that you got earlier, right?

14 A. Right. I don't understand your question.

15 Q. Well, your e-mail is responding to an e-mail, correct?

16 A. Yes.

17 Q. And the e-mail that you're responding to you had to
18 receive before you wrote your response?

19 A. That's how it works, yes.

20 Q. Okay. Good. And in your e-mail responding to the e-mail
21 that you got earlier, you say:

22 "I don't see how you can open Java without
23 Sun since they own the brand and the IP."

24 Do you see that?

25 A. Yes, I do.

1 Q. Now, IP refers to intellectual property, correct?

2 A. That's correct.

3 Q. Patents, copyrights and the like, correct?

4 A. That's correct.

5 Q. And when you say, "I don't see how you can open Java
6 without Sun," you're talking you don't see how you can open
7 Java without Sun's agreement, correct?

8 A. Specifically, what I meant here is --

9 Q. No, I'm asking is that what you meant? That's exactly
10 what I'm asking. That's a yes or no question.

11 A. Can you ask it again, please?

12 Q. Sure. When you say, "I don't see how you can open Java
13 without Sun," what you're saying is that you don't see how you
14 can open Java without Sun's agreement or permission. Fair,
15 sir?

16 A. Can you -- sorry, can you ask it one more time?

17 Q. Sure. When you say, "I don't see how you can open Java
18 without Sun," what you're talking about is you don't see how
19 you can open Java without Sun's agreement or permission.
20 Correct?

21 A. Uhm, yes, that's correct.

22 Q. Okay. Then later on that day -- well, first of all,
23 Mr. Stein then sends you another e-mail, correct?

24 A. Apparently, yes.

25 Q. And then you respond the same day to Mr. Stein's second

1 e-mail, correct?

2 A. Yes, I do.

3 Q. And you begin, "Ha, wish them luck."

4 Do you see that?

5 A. I do.

6 Q. And then you go on to say, "Java.lang APIs are
7 copyrighted."

8 Do you see that?

9 A. I see that.

10 Q. And you meant copyrighted by Sun, correct?

11 A. Uhm, I didn't say exactly that in this e-mail.

12 Q. You meant Sun, did you not, sir?

13 A. Give me a second to think back and I'll think back and
14 figure out what I meant when I typed it.

15 Q. Okay. Take your time.

16 A. In the context of this, I think that meant that the APIs
17 were copyrighted.

18 Q. Copyrighted by Sun?

19 A. Yes.

20 Q. Okay. And, in fact, you then go on to emphasize in the
21 very next sentence:

22 "And Sun gets to say who they license the TCK
23 to."

24 Correct?

25 A. I see that, yes.

1 Q. And then you go on to say that this forces you to take the
2 shared part, which taints any clean-room implementation.

3 Do you see that, sir?

4 A. Yes, I see that.

5 THE COURT: Is this a good breaking point?

6 MR. BOIES: Yes, it is, Your Honor.

7 THE COURT: All right. We have reached the magic
8 hour, 1:00 o'clock. So remember all of the admonitions. We'll
9 see you back here at the normal time. I thank you for everyone
10 being on time so many days in a row. It's wonderful. We'll
11 see you here tomorrow at 7:45. Thank you.

12 THE CLERK: All rise.

13 (Jury out at 1:00 p.m.)

14 THE COURT: All right. Everyone, be seated.

15 Mr. Rubin, you're free to go for now. Please be back
16 at 7:30 in the morning. We'll finish your -- I think we'll
17 finish tomorrow. And you're on, effectively,
18 cross-examination. You still work for Google, right?

19 THE WITNESS: I do, that's correct.

20 THE COURT: All right. So you cannot talk to the
21 lawyers about your testimony. You can talk to them about
22 things like how to get to court and so forth, but no talking
23 about the subject matter of your testimony.

24 THE WITNESS: I understand.

25 THE COURT: All right. Thank you, sir. You're free

1 to go.

2 And, now, if you want to take up these other
3 exhibits, this would be a good time to do it. How do you want
4 to use the time?

5 **MR. BOIES:** Absolutely, Your Honor. Can I ask the
6 Court how far I got before we stopped?

7 **THE COURT:** We got up to 207. We hadn't ruled on
8 that yet.

9 **MR. VAN NEST:** Your Honor?

10 **THE COURT:** Yes.

11 **MR. VAN NEST:** This is sort of a problem we've been
12 discussing over the weekend. But they've been asking -- we've
13 now stipulated -- we Google -- to more than 120 exhibits. And
14 it's one thing if we've got source code or compilations and,
15 you know, we've done that. We've cleared off their need to
16 call several witnesses doing this.

17 Starting over the weekend, we got lists of 90 e-mails
18 that are what I'll call testimonial-type things, where, you
19 know, this is an e-mail or it's a presentation. They're
20 clearly Google documents, that's true.

21 But shoveling all of this stuff into the record
22 without a witness and then putting together a slide show for
23 the closing is not what I think Your Honor had in mind.

24 I understood from the other day that you were guiding
25 me and telling me, hey, if they've got some that are clearly,

1 you know, not objectionable and they're from your client and we
2 don't want to have people unreasonably fighting about stuff,
3 yes, I got that.

4 And we have, as I say, now stipulated to more than
5 125 such exhibits, which is far more than, you know, a jury can
6 comprehend in a reasonable closing of whatever length you were
7 to give us. I think you've given us 90 minutes each.

8 Now all we're doing with these is shoveling a whole
9 bunch more. What I've said to Oracle is, you're going to use
10 it with a witness? Fine. They wouldn't even tell me which
11 ones they wanted to use with the witness, so that's okay.

12 But -- if you're going to use it with a witness,
13 great, but I don't want to have to stipulate to 50 or 60
14 documents that are just randomly floating out there in the case
15 if they're never going to be used.

16 So I think it's improper to go through these long
17 lists and throw a whole bunch of e-mails into the record so
18 people can cherrypick them at the end.

19 **MR. BOIES:** Your Honor, these are not randomly
20 selected e-mails. All of the things that we're dealing with
21 right now are documents that were either, in part or in whole,
22 prepared by or sent to Mr. Rubin, the top executive of Android
23 at Google, in connection with the work that they were doing.

24 I think -- we asked them to -- there are 30 -- 30. I
25 don't think there are 90. There are 30 or 40. But these are

1 all documents that relate to this subject matter. They are
2 from or to the single most important executive for this
3 project. And I think that that is not at all an unreasonable
4 number to have in the record.

5 Now, I am prepared, if I have to, to go through the
6 evidentiary foundation with a witness. I think that is a waste
7 of time. These are all documents produced from Google's files
8 bearing Google's document production stamp on their face or to
9 the certain knowledge of counsel or to or from this witness. I
10 don't think there ought to be any objection.

11 I'd hoped to work this out over the weekend. We were
12 not able to do so.

13 **MR. VAN NEST:** My point isn't that the witness isn't
14 going to authenticate or provide foundation. That's not it.
15 That's not it.

16 I got a list of 97, and then it went to 70 and then
17 it went to 40. My point is if it's going to be discussed with
18 a witness and you know you're going to discuss it, fine. I've
19 got no problem.

20 And I'm not insisting that every single document in
21 the case be thoroughly examined for foundation. And we haven't
22 done that. There have been a few where there's been a dispute,
23 and that's the case in any trial. All I'm objecting to is this
24 sort of wholesale shoveling of exhibits into the record in a
25 wholesale fashion.

1 We all know what the key exhibits are. If Mr. Boies
2 tells me this is one I'm going to examine on, fine. But I just
3 don't think that's the way to run things and I don't appreciate
4 being asked to stipulate to 50 at a time so they can be
5 shoveled into the record.

6 **MR. BOIES:** Your Honor, we're not shoveling into the
7 record. This is not something I'm putting 200 documents into
8 the record. Even that wouldn't be unusual in a case of this
9 importance.

10 **THE COURT:** Well, are you going to use each one of
11 these with the witness?

12 **MR. BOIES:** I don't think I will have time to use
13 each one of these with this witness, Your Honor. These are
14 documents which I might very well want to use. I think all of
15 these documents I'm going to want to use, assuming I have the
16 time, with one witness or the other.

17 These happen to be documents that are to or from this
18 particular witness. And I don't want to get caught in a
19 situation in which, when I want to use it with the witness, I
20 get an objection, which I got earlier when I tried to use a
21 document with another witness, that there was no foundation
22 because it wasn't to or from that particular witness.

23 This is -- this is my opportunity, with Mr. Rubin on
24 the stand, to get these documents in. I think I'm entitled to
25 get them in.

1 **THE COURT:** Well, all right. Have you both said what
2 you want to say?

3 **MR. VAN NEST:** I just want to go back to my basic
4 point. This is not the way I'm accustomed to conducting
5 trials. If he's got a handful -- and I've said this a dozen
6 times.

7 **THE COURT:** Wait, finish that thought. He has a
8 handful of what?

9 **MR. VAN NEST:** I told Mr. Jacobs on Saturday, if
10 there's a handful of exhibits that you want to use with the
11 witness and you want a stipulation to move things along, I'm
12 certainly going to consider it, and that's fine.

13 We've been -- we've got 125 of them we've already
14 stipulated to. But shoveling 30 or 40 in at a time when we
15 know darn full well they're not going to be able to use them
16 all with Mr. Rubin is not right.

17 Every one he's been asked about so far he said, yeah,
18 I recognize this, I know what it is. And I expect you're going
19 to find that throughout his examination. And if there's a
20 couple of exceptions to that, you know, so be it. But I just
21 don't think it's --

22 **THE COURT:** Okay. You're repeating yourself.

23 **MR. VAN NEST:** You're right.

24 **THE COURT:** Here's the answer. There are several
25 parts to this answer.

1 First, there's the evidentiary rule; and that is, if
2 you don't want to stipulate, fine. You don't have to
3 stipulate. They can lay the foundation in the normal way.
4 That's point one.

5 Point two, normally the lawyers work this out. And
6 most of these would come in by agreement.

7 Point three, why is this coming up? I know the real
8 reason. It's a time issue. Mr. Boies wants to get all of
9 these in without using up his time. You're up to 686 minutes.
10 They're running short on time. They may run out of gas before
11 the airplane lands at the airport.

12 (Laughter)

13 **THE COURT:** So that's what's really going on here.
14 And you want to put them in that position. You want them to
15 run out of gas before the airplane gets to the airport. So,
16 you know, it's part of the problem of having time -- in other
17 words, but you have this legitimate point.

18 I'm going to -- I'm now -- Mr. Van Nest has a
19 legitimate point. And that is that by letting these come in
20 without some witness explaining them, then some -- in the
21 middle of summation, suddenly you see something on the screen
22 that was an admission that you didn't realize and it's being
23 used in a way that could have been explained if the witness had
24 ever been given that opportunity. And because it was, as you
25 say, shoveled into the record without explanation, suddenly

1 you're in a bad way.

2 I've been in that position. I understand that
3 problem. So here is what -- you have two alternatives. And I
4 commend the first one to both of you. And that is, when
5 there's a long list like this, I think you would be within your
6 rights to say -- I'm picking a number that I think from
7 experience is about right.

8 You pick 25 percent of them and say, these I'm not
9 going to stipulate to, you've got to have a witness identify
10 them. And then the other 75 percent you let them come in by
11 agreement.

12 Now, that would be about right. I think --
13 25 percent, one out of four, are the only problematic ones.
14 There's not much the witness could do to explain those away or
15 not. But maybe you would say these -- these you've given us --
16 you asked us for 40, these 30 we're okay with, these 10 you're
17 going to have to show to a witness.

18 Now, that's the practical way to handle this, but I
19 can't force you to do that because that's not within the rules
20 of evidence. That's just from practical experience. So if you
21 don't want to agree to that, then you are entitled to make
22 Mr. Boies do it the hard way and use up his time.

23 And if you do this, it ought to work both ways.
24 Whenever you give a long list to the other side, they ought to
25 reciprocate, if it's an internal company records.

1 **MR. BOIES:** With that guidance, I think we can --
2 with that guidance, Your Honor, I suspect we'll be able to work
3 it out.

4 **THE COURT:** I think you ought to talk about it and I
5 think most likely you'll agree.

6 **MR. BOIES:** Right.

7 **THE COURT:** Point made.

8 **MR. VAN NEST:** I just want to say one thing, Your
9 Honor. My point that you identified of mine as a legitimate
10 point, that's the only point. It's not a time point for me.
11 I've already stipulated to 125 of these and many more. So it's
12 not a time point for me. It's not.

13 **THE COURT:** I know.

14 **MR. VAN NEST:** It's exactly the point you're making.
15 I don't want to be in a situation --

16 **THE COURT:** All right. Well, okay. From their point
17 of view it's a time point because they're up to 686 minutes.

18 **MR. VAN NEST:** But that's not my problem. That's
19 their problem.

20 **MR. BOIES:** And these are basically one and two-page
21 documents. It's not like there's something that's hidden there
22 that they're not going to know about that's going to spring
23 out.

24 **THE COURT:** Usually that's right. I agree. Out of
25 10 documents, there's probably one there that's subject to some

1 kind of -- well, I'm saying 25 percent should hold everything.

2 If you want to hold back your consent on 25 percent,
3 that would be entirely reasonable. So if they give you 40, you
4 hold back on ten. If they give you a hundred, you hold back on
5 25.

6 **MR. VAN NEST:** The reason I used handful, Your Honor,
7 is it's a little bit easier to see what a handful is. If you
8 say 25 percent and I get a list of 200 exhibits tonight, I
9 can't wait, that's probably going to be it --

10 **THE COURT:** Well, that would be unreasonable. In
11 fact, 40 is a lot. I would not have asked the other side to
12 stipulate to 40. I'm thinking 12, maybe two dozen. For
13 documents in that range, numbers in that range is manageable.

14 But 40, I think you're giving a big homework
15 assignment to the other side. When they're trying to do their
16 cross-examination, I think that's a lot. I wouldn't have asked
17 for 40.

18 **MR. BOIES:** But I'm greedy, Your Honor.

19 **THE COURT:** I can't say never. I can just say it
20 seems like a lot.

21 **MR. BOIES:** It does, but this is a very important
22 witness, Your Honor.

23 **THE COURT:** Of course. You could have asked before
24 last night.

25 **MR. BOIES:** Well, actually, all these documents were

1 in what we gave them on Saturday. And we'll go over them again
2 and see what we've got.

3 **THE COURT:** All right. You all work on that.

4 I've got some questions for you though. Can I change
5 the subject?

6 **MR. VAN NEST:** Sure.

7 **THE COURT:** Let me just give you the numbers. 686
8 and 437 are how I calculate the time so far. And I think under
9 the hour each side gets 1,020.

10 Is there -- on the willfulness issue, is there going
11 to be a defense of advice of counsel?

12 **MR. VAN NEST:** No, Your Honor.

13 **THE COURT:** Well, I have this question then. I'm
14 working up a verdict form. And do you want -- I told you
15 earlier we would save willfulness until the end. That's
16 usually just because of the defense advice of counsel.

17 So if both sides were to agree to advance the issues
18 of willfulness to this phase, we could have the jury -- you
19 don't have to answer that now. I want you to think about it,
20 because it's on my mind.

21 And, you know, we can stick with the way we had it, I
22 guess, but if there's not going to be advice of counsel, then
23 the jury is hearing all these documents about we need a license
24 and so forth, it does go to willfulness.

25 So if the jury finds that you did something wrong,

1 you know, this might be a logical time to figure out if it was
2 willful or not. So I'll let you consider that point.

3 I have a different question, and that is this -- this
4 is to Oracle. Here's something I have a hard time following,
5 and that is your argument about -- what's the phrase? --
6 derived work?

7 **MR. BOIES:** Derivative work.

8 **THE COURT:** Derivative work. Derivative work.

9 Your argument is that if somebody starts with the
10 plain English descriptions in a clean room and then they do --
11 you know, they put on their own scientific thinking caps and
12 come up with their own program code, that no matter what they
13 come up with is going to be an infringement, a derivative work.

14 Doesn't that argument violate the principle that no
15 one has a monopoly or ownership of the idea expressed? They
16 have the -- they can -- they have the right to the expression
17 of their particularized expression of the idea, but if you
18 were -- say how do you do the rangeCheck or how do you do
19 the -- find the maximum of two numbers, and that that is the
20 idea and then you let a bunch of students go wild on it and
21 each one of them come up with a somewhat different solution to
22 that problem, isn't that the classic example of the -- you
23 don't have the right to their expression of that idea?

24 Now, possibly I'm misunderstanding your derived work
25 argument, but isn't that right? I mean, that just seems to me

1 to be invalid under the basic tenets of copyright law.

2 I'm just talking now about the derived work part.

3 That is all. Using the plain language specification and then
4 saying anything in the world, no matter what they came up with,
5 has to be a derived work based on having started from the plain
6 English explanation of what that module is going to do.

7 Mr. Boies.

8 **MR. BOIES:** Your Honor, I'm not sure this is going to
9 be terribly helpful, but I think it depends on whether you view
10 what you refer to as the plain English version as being simply
11 an idea or actually being creative expression.

12 If it is a creative expression and they are copying
13 that and interpreting that and deriving what they have created
14 from that, then it is a derivative work. If it is merely an
15 idea, then I think the Court is probably right. And I think
16 that the issue here is which side of that line does this fall
17 on?

18 And the Court will note that everybody -- before this
19 litigation, everybody on both sides thought that a clean room
20 operation didn't save you from getting a license.

21 **MR. BOIES:** You needed to both have a clean room and
22 a license. All the clean room did was it permitted you to
23 develop your own code. You didn't have to license the code
24 from Sun or Oracle, but a clean room did not excuse you from
25 getting the specification license.

1 **THE COURT:** That's not -- I mean, that's what that
2 particular memo said. Yes, I grant you that, but I'm not sure
3 that's required under the law.

4 That's, you know, a good argument to the jury and
5 you'll be most effective in making it, but nonetheless I don't
6 think that's the law that -- I don't think Mr. Rubin and Google
7 ought to be held to, as a matter of law, to something somebody
8 writes in an email if it's incorrect under the law.

9 Why do these companies have clean rooms to begin
10 with?

11 **MR. BOIES:** Well, remember, it wasn't just an email.
12 I mean, that's what they said in every one of their
13 presentations that addressed the issue. They said it in a
14 variety of emails.

15 That's why Danger took the license. They weren't
16 using the code. They were going to have a clean room
17 implementation --

18 **THE COURT:** But were they using -- I'm sorry. Say
19 that part again. Danger was what?

20 **MR. BOIES:** Danger took a license even though they
21 were not going to use the code. And the reason they took the
22 license was because even if you've got a clean room in which
23 you create your own code, the specification of what you're
24 developing that comes from Sun, and now Oracle, is something
25 that causes you to need a license. That's the derivative work

1 argument -- or that is a derivative work argument.

2 **THE COURT:** You see, but didn't it cover the
3 trademark, too? Didn't the Danger thing cover the trademark?

4 **MR. BOIES:** I suspect it did, your Honor, but it
5 clearly -- it clearly covered a specifications license.

6 In other words, it clearly thought they needed a
7 specifications license and when they filed their SEC statement,
8 that's the way they described it.

9 **THE COURT:** All right. Well, I still question the
10 premise.

11 Now, it may be that they agreed on a -- as a
12 concession to the shortness of life, that they agreed to a
13 license that required a specification, just in the same way as
14 I'm sure both companies here have agreed to license some patent
15 from a troll who had a completely invalid patent, but it wasn't
16 worth fighting over it.

17 I mean, the fact that somebody makes an agreement
18 doesn't necessarily mean that the law required it.

19 **MR. BOIES:** I think that's fair entirely, your Honor.

20 **THE COURT:** So I want to come back to my basic
21 question, which is: If you have a document in plain English
22 that says this particular class or method -- let's say method.
23 This particular method will return the larger of two numbers,
24 and then you gave that -- I am sure if you went and looked in
25 textbooks, you would find examples of that very exercise,

1 teaching young people in college how to perform that writing
2 various kinds of code.

3 And to say that you own every single one of those
4 implementations just because you came up with the idea to
5 compare two numbers, I think that's the classic case of
6 overreaching, of trying to get a copyright on every single
7 expression of an idea.

8 **MR. BOIES:** Your Honor, that is not what we're --
9 we're not trying to do that. And I'm probably not explaining
10 as well as I should --

11 **THE COURT:** I know you have a lot of -- I'm ignoring
12 the Structure, Sequence and Organization point.

13 My job is to come up with a list of questions to give
14 to the jury that is actually just as clear as possible and gets
15 to the issues that are actually in play in the case. I don't
16 want them to have to take time over something if it's invalid
17 as a matter of law.

18 I'm going to point to one on the other side in a
19 minute that I want you to consider.

20 I'm just asking you to consider, is that one even
21 worth taking to the jury.

22 **MR. JACOBS:** Your Honor, I think the question needs
23 to be reframed in light of what we're alleging.

24 **THE COURT:** Yes.

25 **MR. JACOBS:** It is not the taking of any idea for any

1 particular method that is -- which is the hypothetical that the
2 Court just posed.

3 Surely if you say to a coder, "I want you to return
4 the max value," and he goes and writes code to return the max
5 value on his own, that's not copyright infringement.

6 What we're -- that's why this -- similarly, if you
7 said to a movie maker, make a movie about an adventure seeking
8 the Holy Grail, and independently the movie maker went off and
9 made a movie about seeking the Holy Grail, that would not
10 infringe the copyrights in *Monty Python and the Holy Grail*
11 Umberto Eco's *Holy Grail Seeking* book.

12 **THE COURT:** Or *Indiana Jones*.

13 **MR. JACOBS:** Exactly. And so you can't look at it at
14 that level of idea. You have to look at it at this level.

15 (Demonstrative displayed.)

16 **MR. JACOBS:** And you have to ask yourself, if you
17 turn this chart and all of the stuff that's not on the chart,
18 the methods, et cetera, that aren't even shown here and you
19 turn that over to your coders, well, what's the analogy there?

20 The analogy there is turning over an incredibly
21 detailed --

22 **THE COURT:** But that's your Structure, Sequence and
23 Organization argument.

24 **MR. JACOBS:** But it is still -- I think we -- it is
25 nonetheless the case that if you turn a script like this over

1 to a movie maker and he makes a movie, that movie is a
2 derivative work of this movie script.

3 The derivative work theory really doesn't arise at
4 the level of what is copyrightable or not. The derivative work
5 theory is in the case merely because they have taken, by their
6 account, something that is in the form of web pages and
7 documentation and converted it into another form of program
8 code.

9 **THE COURT:** Let me try my question a different way.

10 Mr. Baber, I see you're standing up back there, but
11 not yet.

12 Let's say we do ask the jury two questions that did
13 not surface yet. Let's say we ask the jury the question: With
14 respect to these 37 packages, did Google copy the Structure,
15 Sequence and Organization? Question one. And that's at the
16 implementing code level.

17 And then same question, but for the documentation.
18 Did they copy the same Structure, Sequence and Organization at
19 the plain English level, documentation level?

20 So let's assume we're going to ask those questions to
21 the jury, and you'll either get a "yes" or "no" answer on that.

22 So then my next question -- then what good is it,
23 what do we add to that if you've already gotten a "yes" on
24 those or if you've already gotten two "nos" on those.

25 What does it add to then say: By the way, is the

1 implementation a derivative work of the plain English statement
2 of what the method will do? To my mind there is no incremental
3 legal significance at all of that third question. Because if
4 you win on the first two, you're okay. If you lose on the
5 first two, what possible -- the structure sequence has been
6 rejected by the jury. I don't see how they can return a
7 meaningful verdict on derivative work.

8 **MR. JACOBS:** I think that's right, your Honor. If
9 the -- if the protectable material in the Application Program
10 Interfaces is copied in any way, whether it's copied into the
11 specification or into the code -- and I think that's really
12 what you're asking them to decide with the first and second
13 questions -- then you've effectively answered the derivative
14 work question.

15 **THE COURT:** Well, no, no. Because I may have
16 misunderstood.

17 On the second question it would be -- the first
18 question is: Compilable code to compilable code, is the SSO
19 infringed?

20 The second question is: Plain English to plain
21 English, at that level is it infringed.

22 But what I don't see as adding anything, except
23 violating the principle of you can't get a monopoly and
24 ownership over an idea, is that documentation down to -- if you
25 lose on the first two, I mean, I don't see how you can possibly

1 win on the third. And if you win on the first ones, then I
2 don't see what it adds to add that derivative work thing.

3 **MR. JACOBS:** Let me -- let us consider this, your
4 Honor.

5 **THE COURT:** Would you do that?

6 **MR. JACOBS:** I would like to think hard about it.

7 **THE COURT:** Now, here is the similar type of
8 heartburn I want to give to the other side. And I'm not making
9 a final ruling on any of these items, but I see your diminimus
10 argument as -- I don't see how you can say it would be
11 diminimus to copy this Structure, Sequence and Organization
12 because I'm going on the assumption that that is a protectable
13 element. That's what I'm going to say to the jury.

14 It is protectable, but I'm reserving that could be
15 that after the trial is over I'll say, no, it's not. The
16 reason I'm possibly would flip flop on that is that it's
17 important to tee that issue up for the jury so that if the --
18 if I were to rule for Google on the protectability argument, at
19 least the Federal Circuit could reinstate the verdict and we
20 don't have to come back here and do this trial all over again.

21 And that also gives them an opportunity, the jury, to
22 rule on your fair use argument. It may be that the jury says,
23 yes, you did copy it, but, no, it was all fair use. So there
24 are a lot of reasons to do it that way.

25 But here is my main point for you. If it is -- if

1 SSO is protectable under the Copyright Act, isn't it -- no
2 reasonable jury could find that that is diminimus, could they?
3 I find that a hard sell.

4 So here comes Mr. Baber who is the master of hard
5 sells, and he's going to contradict me.

6 (Laughter.)

7 **MR. BABER:** Well, your Honor, I'm happy to answer
8 your question, but I would also like to respond to the exchange
9 with Mr. Boies and Mr. Jacobs.

10 **THE COURT:** Well, you can do that. But answer my
11 question and then you can go back to them.

12 **MR. BABER:** Your Honor, I think that a jury could
13 find that the SSO, just the SSO of the 37 packages is
14 diminimus, and here is why.

15 We continue to believe very strongly -- and we
16 briefed it yesterday and, frankly, we do need to do another
17 brief based on what they said yesterday. But to the extent
18 there is something that is the SSO that is separate and
19 independent from the names, the organization that is provided
20 by the language that puts these things into packages and
21 classes and all of the code that implements it, if there is
22 something that at the end of the day can be identified as the
23 SSO of just these 37 packages -- okay, that's what we're
24 talking about, some separately identifiable thing called just
25 the SSO for just these packages. If we can find that, then we

1 believe that that is an extraordinarily small element of both
2 the Java Platform and of Android.

3 These 37 packages in Java, there's only 37 out of
4 166. And those 166 will have this separately identifiable SSO,
5 whatever it is, but it's also going to have a huge amount of
6 code. It's going to have all these names. And you've said
7 they're not protectable. The names go out. So we can't be
8 looking at the names. That's not part of the SSO.

9 And at the end of the day what you're left with is
10 something in the way of some skeletal thing having to do with
11 the arrangement of this group of packages within a work that is
12 this big. You've got not just 166 API packages, with all the
13 code that implements it and all the API specs, you have got a
14 Java virtual machine. You have got a Java compiler. You have
15 got a Java software development kit. That's their work.

16 This goes to the works as a whole issue. But in the
17 context of that work as a whole, if all we're talking about is
18 the SSO of this little group of the packages, yes, your Honor,
19 I believe a jury could find that that's diminimus.

20 **THE COURT:** Let's say we let the jury decide fair
21 use. Isn't the extent to which something is used a factor
22 there?

23 **MR. BABER:** Absolutely.

24 **THE COURT:** So what more does it add to put diminimus
25 in there?

1 **MR. BABER:** Because diminimus -- and I want to
2 clarify for the Court, just so it's clear.

3 But the diminimus arguments we made, those we made
4 them primarily with respect to these 12 files or the nine lines
5 of code --

6 **THE COURT:** All right. So would you agree to this:
7 That you limit your diminimus argument to those 12 files?

8 **MR. BABER:** Your Honor, I would like to confer with
9 the client about that, but that would be a much harder sell.

10 **THE COURT:** What?

11 **MR. BABER:** Trying to make the diminimus argument as
12 to the 37 would clearly be, that's a whole nother type of hard
13 sell. Okay?

14 **THE COURT:** You agree that would be a hard sell.

15 **MR. BABER:** I agree it would be, certainly, a far
16 harder sell.

17 **THE COURT:** Very close to my position then.

18 **MR. BABER:** I'm trying, your Honor.

19 **THE COURT:** I think what you ought to do is talk to
20 your client and see if we can't streamline the issues that
21 would go to the jury on both sides. I think there is some
22 value. And maybe you both come up with a way to concede a
23 point and keep your main paints.

24 All right. Now, what do you want to say on the
25 other --

1 **MR. BABER:** Just to finish that, your Honor. It
2 really does depend on where you come out on this work as a
3 whole issue.

4 The SSO of the 37 is one thing. If you're saying,
5 hey, they can just say these 37 are a work. Well, if it's just
6 the SSO of those 37, that diminimus, I probably can't even sell
7 that one.

8 **THE COURT:** That one is a closer call and I'm not
9 going to ask anyone to do that.

10 **MR. BABER:** Okay.

11 **THE COURT:** You wanted to respond to Mr. Jacobs?

12 **MR. BABER:** Yes, your Honor. Very briefly I will.

13 There is a great analogy I think. What the API specs
14 do is they basically say, Okay, programmers, and app developers
15 and everybody else in the Java world. In Java prewritten code
16 that performs this function has this name. That's what the API
17 specs do. It's like a dictionary. It says -- it goes both
18 ways. If you want code that does X, here is the name you call.
19 And if you see this name, this tells you what that code does.
20 Just like a dictionary.

21 So this is not like providing a script for a movie or
22 anything like that. The analogy is, this is like giving
23 somebody a dictionary and saying, "These are all the words you
24 can use in this programming language. Now, with all these
25 words, can you go out and write some great programs?" It's not

1 like giving somebody a script and saying, "Here is the script.
2 Now make the movie from the script." That would be like
3 saying, "Mr. Movie Maker, here is a dictionary. Can you go out
4 and write a really good script using the words in this
5 dictionary?" That's the analogy.

6 **THE COURT:** Well, but you're assuming that a
7 programming language is not -- even a very advanced one that
8 has prewritten modules, is not copyrightable. That's one of
9 the issues here.

10 **MR. BABER:** I don't think so, your Honor. I think
11 the issue -- there's two issues here in this question.

12 One is names. You have already ruled on that. The
13 names are out of the case. So what the names are --

14 **THE COURT:** Where do you -- we had that declaration.
15 Put the chart back up there. Tell me what you think -- the one
16 that the witness the other day did?

17 (Chart displayed)

18 **THE COURT:** Which part of that do you think is the
19 name?

20 **MR. BABER:** I think the name, your Honor, Dr. Bloch
21 testified that each method, for example, has under the language
22 spec a fully qualified name. In this case, the name is --
23 java.lang.Math.max. And you said in your summary judgment
24 order that the names that are unprotectable, the names and
25 short phrases that are not protectable, include the package

1 names, java.lang; the class names, java.lang.Math; and the
2 method names.

3 **THE COURT:** All right. Isn't that what I said?

4 **MR. JACOBS:** I don't think it was that clear.

5 And you also said -- and this is the issue we briefed
6 to you -- that the selection, range and coordination of names
7 was a reserved issue.

8 **THE COURT:** Say that part? What was reserved?

9 **MR. JACOBS:** The selection arrangement coordination.
10 Not any one of these names, your Honor -- not any five of those
11 names. Thousands of them, selected and arranged and organized
12 and structured in sequence by the Java programmers. They
13 cannot be that merely because we have attached the name Name to
14 any one of these elements, this is not protectable.

15 And if you want to go with an even better analogy, or
16 different analogy, I would go to the West Keynote System where
17 it is structured and arranged. And, of course, any one name of
18 a keynote is not protectable. But if you took -- if you gave a
19 programmer the West Keynote System and you said, turn that into
20 programming, of course, that would be infringing.

21 **THE COURT:** Well, wait a minute. How do you know?
22 You said "of course."

23 **MR. JACOBS:** Because they -- the circuits are split
24 on whether the pagination is protectable, much less the whole
25 structure of the taxonomy.

1 **THE COURT:** Well, where does it -- has the circuit
2 already held that West has a copyright on that key system?

3 **MR. JACOBS:** There is no case on the key system. No
4 one would take a detailed schema like that and say you can just
5 copy it and create Deerings copying West Keynote System, even
6 though we all use those key notes when we write legal
7 memoranda. That is a detailed taxonomy that was created by
8 West to organize all this material.

9 **THE COURT:** All right. What do you say to that one,
10 Mr. Baber?

11 **MR. BABER:** Your Honor, I don't think that's what
12 we're dealing with here because the West taxonomy has content.
13 It's not just defining words like these things do.

14 These things, in effect, make up new words that
15 haven't existed before in the Java language. When I write a
16 new API package, let's call it java.bruce, okay? And it's got
17 to have some new function that's never before been in the Java
18 language.

19 Well, I'm now creating a whole new bunch of
20 vocabulary words. West never did that. West takes the normal
21 English language and uses them in the normal English language
22 way. But once I define java.bruce -- and I might have
23 Java.bruce.History, java.bruce.Economics, java.bruce.Whatever.
24 I've now made up a bunch of terms and anybody that wants to
25 have their Java program recognize those terms and use them in

1 the same way I've defined them, has to call them the same
2 thing, put them in the same place, have them be organized the
3 same way as to packages and classes, and it's become a part of
4 the language.

5 **THE COURT:** But those long, extended names,
6 Mr. Jacobs' point is that they come with a system of
7 organization. So that -- and it's not just randomly --
8 randomly selected words with dots in between. Each one, they
9 bear a relationship among each other in a hierarchal way so --

10 **MR. BABER:** Full stop. Full stop, your Honor, if I
11 may.

12 The hierarchal way that they are talking about,
13 that's what the language requires. The language says they have
14 to be that way. It only looks like hierarchy to us humans
15 because we need to be able to see it in some form we can
16 understand it, but to the computer it's all text and it's all
17 just names.

18 You can have java.lang.Math.max and right after it
19 you can have java.arrays.java. They are not in alphabetical
20 order. They are not in a form that any human would see them as
21 organized.

22 The organization is simply a human artifact that is
23 pulled out of the code through a program to make it look like
24 something we recognize, like a table of contents for a book.
25 So in the code itself all you have is one line after another

1 after another of code.

2 **THE COURT:** Let me ask a different question to
3 Oracle.

4 Are you -- I think I've asked this before, but are
5 you saying that each -- it's your analogy to the magazine. You
6 used that the other day, or maybe this morning.

7 You said each of these 37 APIs are like 37
8 advertisements in a magazine. So are you saying that each one
9 of these is a separate stand-alone thing, or are you saying
10 that the SSO of the entire 37 is the protected item?

11 **MR. JACOBS:** I think we're saying both. I think the
12 37 packages as a group and each package.

13 So a package is a huge -- we already know there's
14 lots of classes in a package. It has a Structure, Sequence and
15 Organization to it. It's got a coherence to it as the highest
16 level of the organizational structure.

17 So aside from the platform itself, that we're at the
18 package level. And then the 37 that they chose to took, also
19 define a body. And I think we win under both theories and I
20 think both are valid.

21 **MR. BABER:** May I, your Honor?

22 **THE COURT:** Sure.

23 **MR. BABER:** I don't see how there could possibly be a
24 principled way of saying that you can -- well, let me back up
25 to say, we disagree that the work should be anything other than

1 the whole work.

2 But if you want to talk about some part of the work
3 as to which you can talk about SSO, I can't imagine any
4 principled way in which they could say it would be anything
5 less than the SSO of all their APIs, all 166 packages, because
6 they all work together.

7 You've heard the testimony now from Professor
8 Mitchell and from Dr. Reinhold and from Dr. Bloch that every
9 package, every class has to come back to java.lang. Because
10 java.lang is what tells you, defines a class. What's a class?
11 What's an object? What's a string?

12 So to the extent there is any chance of an
13 identifiable smaller unit within the Java Platform that has to
14 do with these APIs, the SSO, they have to be talking about at
15 least all the APIs because they all have the same SSO and they
16 are all intertwined --

17 **THE COURT:** But they are only alleging 37. I don't
18 see why you're saying I have to --

19 **MR. BABER:** Because, your Honor, it's the issue of
20 they can't redefine their work and cut it up with scissors to
21 say that only the part that's at issue is the part that's a
22 separate work.

23 **THE COURT:** But if it was a -- that's what the
24 *Hustler* magazine case allowed the plaintiff to do it in that
25 case, where they -- it was a single advertisement in one

1 magazine.

2 **MR. BABER:** Yes, your Honor, and that single
3 advertisement was a stand-alone piece.

4 Mr. Jacobs or someone referenced this morning the
5 Texaco case involving the scientific journals where you have an
6 issue of a journal where there might be 10 articles all
7 together, and the Court said, yeah, we're going to say you can
8 look at just that one article. That, your Honor, I would
9 submit is completely different from what we have here.

10 The articles are separate and stand-alone units with
11 separate authors that can be separately published and you look
12 at it, it's a complete contained work.

13 That's APIs, they have testified, all their witnesses
14 have talked about how interrelated and interconnected this
15 whole big API of all 166 packages is. We think the evidence
16 will show that. They focused on the 37. You've heard some
17 evidence already about some of the others in the group.

18 But by the time the evidence is closed, I think
19 you'll understand that all -- all of these APIs all relate to
20 each other because they all have the same kinds of
21 interdependencies. It's not just these 37.

22 The only reason we have 37 -- and we talked about
23 this this morning -- is we have 51 total in Android that are
24 our Java packages, but they have gotten rid of 14 because they
25 realized they don't own them. There is no magic to 37. It's

1 just because they realized during discovery, when we pointed
2 out to them, that they had to jettison some.

3 So no one has ever, ever before this trial suggested
4 that these 37 are a unit of any kind. They are a unit only
5 because it's what we have, less the ones they don't even own in
6 the first place.

7 **THE COURT:** Mr. Van Nest --

8 **MR. VAN NEST:** Mr. Baber just said what I was going
9 to remind the Court, which is that in a journal where there are
10 articles, the articles have a separates existence.

11 Before this case started, no one ever talked about
12 the SSO of 37 APIs or 51 APIs or any particular number. The
13 only reason it's here is that's what Android is using and they
14 are creating -- that's where it came from, Mr. Baber just said
15 that.

16 Before this lawsuit started, nobody ever spoke about,
17 knew about, talked about, speculated about or argued about some
18 SSO of 37 APIs, or 51, or any.

19 **THE COURT:** I have seen in the e-mails there were
20 some references even by Mr. Rubin to APIs.

21 No, I heard that. That's in the e-mails.

22 **MR. VAN NEST:** He'll be discussing APIs and what he
23 meant in the e-mail, and whatnot.

24 But not even in that e-mail, or anything you have
25 seen, or anything you will see, has anybody been saying, oh,

1 there's an SSO of 35 APIs, and you're using that organization,
2 or there's an SSO of 51, and you're using that, or of a
3 hundred, or 150, or 160.

4 These numbers that you're hearing are a product, a
5 lawyer-made-up product of this lawsuit. They don't exist
6 anywhere else. No one has ever talked about them, discussed
7 them. They are not written down anywhere. They are not
8 published anywhere. They are not fixed anywhere.

9 They're only here because that's what Android uses.
10 And so they want you to define this as the smallest possible
11 unit they can, to have the closest comparison to what's in
12 Android.

13 That's it. There's nothing out there. You won't see
14 it. You haven't seen it yet, and they've got, you know, a few
15 minutes left in their case.

16 **THE COURT:** Well, but -- wait, wait.

17 But here's the bugaboo in that one, it seems to me,
18 for Google. And that is, you chose consciously -- if these
19 APIs weren't so great and they didn't add something and didn't
20 have some coherence to them, why did you choose to include them
21 in Android?

22 I mean, that's kind of a fundamental -- doesn't that
23 sort of prove right there that they have some coherence?

24 **MR. VAN NEST:** Number one, number one, there were 51
25 of them. Right?

1 **THE COURT:** Right.

2 **MR. VAN NEST:** So there's never been a unit of 37.

3 **THE COURT:** All right. It's 51 minus 14. But you
4 did use that number, 37, that happens to be the number that
5 Google itself chose to incorporate because, as you said this
6 morning, the developer community would expect to see them
7 there.

8 **MR. VAN NEST:** As we have said, Your Honor, those
9 APIs were integrated into a much bigger system, which all
10 operates together.

11 And, believe me, we haven't even started our case
12 yet, so I think you are going to get plenty more testimony
13 about this, that our core libraries and our other Android APIs
14 all work together as part of one big system.

15 So, again, it isn't as though there's a unit out
16 there where somebody has identified these 37 APIs and said,
17 hey, they are separately available for sale or have a separate
18 existence, or anything like that.

19 That's my point, is that there is no such unit, other
20 than and apart from this lawsuit. There is no unit out there.

21 **MR. BABER:** Your Honor, may I just add to what
22 Mr. Van Nest said?

23 I understood your question to be saying, well, why do
24 you have Java APIs in the first place, at all? I think that's
25 what you're asking. The answer is: In order to be able to use

1 the freely available programming language.

2 You have now heard testimony from witness after
3 witness after witness, you have to have some of these APIs to
4 use the language.

5 **THE COURT:** Four.

6 **MR. BABER:** Well, four packages, which is hundreds of
7 methods. And then there was testimony from other witnesses
8 that would get into some other classes. So there are clearly
9 fact issues about how much is essential, but at least those
10 four core packages.

11 In order to be able to allow developers to write
12 applications in the Java programming language that anyone can
13 use, we had to have at least some APIs. So we started with
14 what was out there in the Java world. And, as we discussed
15 this morning, we threw out a bunch because they just wouldn't
16 work with Android.

17 Android is a whole nother kind of device that's never
18 existed before in the Java world. We had to write our own
19 APIs. And then with the ones that were left, the ones that
20 developers expected to have to write in the language, we kept
21 them. That was the 51.

22 But to answer your core question, the reason why any
23 of these are in there at all is simply to support and allow
24 programmers to use the Java programming language.

25 **THE COURT:** Is that a fair use argument? Is that

1 what your argument is --

2 **MR. BABER:** Absolutely, Your Honor. It's a fair use
3 argument, especially because all we use are the unprotectable
4 names, method signatures, and ideas that are reflected in the
5 API specs. We don't use their code. We don't use their
6 expression.

7 To the extent we use anything other than the names,
8 to the extent there is any kind of a SSO that's not reflected
9 in the names themselves, we used the minimum amount necessary
10 in order to use those APIs to support the language. We just
11 used the names, the signatures. All the rest of it we did on
12 our own.

13 **THE COURT:** The burden on the fair use is on you?

14 **MR. BABER:** It is absolutely on us, Your Honor.

15 **MR. JACOBS:** Two quick points.

16 **THE COURT:** Yes, Mr. Jacobs.

17 **MR. JACOBS:** I think the argument you just heard
18 actually reinforces our point. There were originally 51
19 packages. But these packages come from a JSR. You heard the
20 testimony on that. A JSR is what launches the development of a
21 package. So at the package level, we see a JSR launching the
22 development.

23 And the package can have -- the 37 here are the 37
24 that Oracle owns. And there's a 38th that Oracle doesn't own,
25 and a 39th that Oracle doesn't own.

1 So at the package level, these are akin to a magazine
2 article or to an advertisement.

3 Second point, what these cases really tell us,
4 whether it's *Harper & Row*, or *Hustler*, or American Geophysical
5 against Texaco, copyright law is pretty protective of the
6 copyright owner. They don't allow the defendant to bury its
7 copying in a larger body of work and say look how trivial and
8 small it is. Copyright law protects against exactly that kind
9 of copying.

10 And so what the jury should be told here, the jury
11 should be certainly told -- however we end up framing this --
12 focus on that which was taken, and its substantiality, not that
13 which was not taken.

14 And the clearest way to do that is to define a body
15 of the work that was taken and ask if that is substantial. If
16 they want to argue that it's not substantial, they shouldn't be
17 allowed to do that by embedding it in a huge, larger project.
18 That's what those cases tell us.

19 **THE COURT:** Okay. We'll leave it there for today.
20 Anything else you need to bring up with the Court before we
21 adjourn for the day?

22 **MR. JACOBS:** Nothing from us, Your Honor.

23 **MR. VAN NEST:** I don't think so, Your Honor. We may
24 need to do some witness interrupting tomorrow, but we'll work
25 that out tonight, given availability.

1 **THE COURT:** Thank you. See you at 7:30.

2 (At 1:52 p.m. p.m. the proceedings were adjourned
3 until Tuesday, April 24, 2012, at 7:30 a.m.)

4 - - - - -
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

E X H I B I T STRIAL EXHIBITSIDENVOL.EVIDVOL.

14			1341	6
15			1341	6
15			1342	6
18			1342	6
22			1342	6
26, 104, 157, 233, 234,			1152	6
29			1343	6
34, 165, 383, 384, 416			1153	6
131			1343	6
140			1343	6
147			1343	6
151			1343	6
154			1343	6
158			1344	6
161			1344	6
165			1344	6
180			1344	6
183			1345	6
203			1345	6
251, 21, 172, 1029, 238			1152	6
612, 617, 2223			1154	6
687			1215	6
729			1153	6
896.1, 896.2, 896.3, 896.4			1153	6
896.5, 896.6, 896.7, 896.8			1153	6
897			1153	6
1026			1162	6
1045			1207	6
1047			1205	6
1072			1335	6
2564			1271	6

- - -

I N D E XPLAINTIFF'S WITNESSESPAGEVOL.**LEE, BOB**

(PREVIOUSLY SWORN)

1168

6

Direct Examination Resumed by Mr. Jacobs

1168

6

Cross Examination by Mr. Baber

1176

6

Redirect Examination by Mr. Jacobs

1203

6

Recross Examination By Mr. Baber

1210

6

MITCHELL, JOHN

(SWORN)

1213

6

Direct Examination by Mr. Jacobs

1214

6

Cross Examination by Mr. Van Nest

1265

6

Redirect Examination by Mr. Jacobs

1329

6

RUBIN, ANDY

(SWORN)

1338

6

Direct Examination by Mr. Boies

1339

6

Under Seal Proceedings

1292

6

- - -

CERTIFICATE OF REPORTERS

We, KATHERINE POWELL SULLIVAN and DEBRA L. PAS,
Official Reporters for the United States Court, Northern
District of California, hereby certify that the foregoing
proceedings in C 10-3561 WHA, **Oracle America, Inc., vs. Google,
Inc.**, were reported by us, certified shorthand reporters, and
were thereafter transcribed under our direction into
typewriting; that the foregoing is a full, complete and true
record of said proceedings at the time of filing.

/s/ Katherine Powell Sullivan

Katherine Powell Sullivan, CSR #5812, RPR, CRR
U.S. Court Reporter

/s/ Debra L. Pas

Debra L. Pas, CSR #11916, RMR CRR

Tuesday, April 23, 2012

*Katherine Powell Sullivan, CSR, CRR, RPR
Debra L. Pas, CSR, CRR, RMR
Official Reporters - US District Court - 415-794-6659*